NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

AN ALGORITHM FOR ALLOCATING ARTILLERY
SUPPORT IN THE AIRLAND RESEARCH MODEL

by

John M. Geddes, Jr.

September 1986

Thesis Advisor: Samuel H. Parry

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS
UNCLASSIFIED	

2a. SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT
	Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	

4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION	6b OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
Naval Postgraduate School	Code 55	Naval Postgraduate School

6c. ADDRESS (City, State, and ZIP Code)	7b. ADDRESS (City, State, and ZIP Code)
Monterey, California 93943-5000	Monterey, California 93943-5000

8a NAME OF FUNDING / SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS			
	PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)

AN ALGORITHM FOR ALLOCATING ARTILLERY SUPPORT IN THE AIRLAND RESEARCH MODEL

12 PERSONAL AUTHOR(S)
Geddes, John M., Jr.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM _____ TO _____	1986 September	69

16 SUPPLEMENTARY NOTATION

17	COSATI CODES		18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	AirLand Battle, AirLand Research Model, Allocation of Artillery, Differential Game, Generalized Value System

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This thesis extends the development of algorithms for modeling planning processes in the AirLand Research Model (ALARM), an on-going research effort at the Naval Postgraduate School. An algorithm is developed to determine optimal mission assignments for supporting combat resources based on the determination of optimal firer-target combinations. The method of differential games is adopted as the optimizer for the algorithm. The algorithm is applied to a problem of determining artillery battalion mission assignments in supporting a U. S. brigade engaged with an enemy division. The algorithm is solved using FORTRAN 77 and the IMSL routine DGEAR.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (Include Area Code)	22c OFFICE SYMBOL
Samuel H. Parry	(408) 646-2779	Code 55Py

DD FORM 1473, 84 MAR   83 APR edition may be used until exhausted   SECURITY CLASSIFICATION OF THIS PAGE
All other editions are obsolete

An Algorithm for Allocating Artillery
Support in the AirLand Research Model

by

John M. Geddes, Jr.
Captain, United States Army
B.S., United States Military Academy, 1976.

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1986

ABSTRACT

This thesis extends the development of algorithms for modeling planning processes in the AirLand Research Model (ALARM), an on-going research effort at the Naval Postgraduate School. An algorithm is developed to determine optimal mission assignments for supporting combat resources based on the determination of optimal firer-target combinations. The method of differential games is adopted as the optimizer for the algorithm. The algorithm is applied to a problem of determining artillery battalion mission assignments in supporting a U.S. brigade engaged with an enemy division. The algorithm is solved using FORTRAN 77 and the IMSL routine DGEAR.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A.   BACKGROUND OF THE PROBLEM

In response to changing technology and political climate, the U. S. Army has adopted a method of warfighting called AirLand Battle Doctrine. This doctrine describes the next battlefield to have indistinct battle lines and intense firepower [Ref. 1:pp. 1-1,1-2]. The division between the front lines and the rear areas will be very blurred as forces penetrate the forward edge of their opponent's defenses and attack units behind the front lines. The doctrine proposes concepts and tenets that will hopefully lead to success by Army units on this battlefield.

One of the basic tenets is depth. The commander must attack the enemy forces not only in front of his forces, but the enemy forces that are supporting or are still not committed [Ref. 1:p. 2-2]. The successful attack of these forces will have benefits beyond that of just destroying the force. Units in rear areas generally are in one of two groups. They may be supporting the forces on the front lines, in which case their destruction will have an impact across a broad front, or they may be in the reserve, as yet uncommitted. The destruction of uncommitted units takes away alternatives of the enemy commander. It follows that the Army must be able to identify those units whose destruction will have the maximum benefit and attack them before others. The attack of these units will disrupt the coherence of the enemy's organization and take the initiative away from him [Ref. 1:p. 2-1].

B.   THE AIRLAND RESEARCH MODEL

As a way of evaluating AirLand Battle doctrine, a model called the AirLand Research Model (ALARM) is under development at the Naval Postgraduate School. The AirLand Research model is an effort to develop new methods of modeling warfare on a large scale, to be used as a tool for evaluating the doctrine of the AirLand Battle. The three primary purposes of ALARM are:

    a.   Develop modeling methodology for very large scale and sparsely populated rear areas.

    b.   Use the methodology in wargaming/simulation with initial emphasis on interdiction.

    c.   Perform research on AirLand Battle concepts. [Ref. 2:p. 2]

ALARM will initially be designed to be a systemic model (i.e., no man-in-the-loop players). This creates the need for decisionmaking algorithms to perform the roles of human players. Eventually, it is anticipated that an implementation with human players will be developed.

The general setting for the initial ALARM model will be the Fifth U.S. Corps area in Central Europe. One reason for this selection is that the general war in the NATO area has been repeatedly studied and there is a strong consensus regarding the outcome of certain 'textbook' scenarios. Secondly, while there is severe doubt that war would ever occur in that area, little doubt exists that a war there would have a major impact on the future shape of the world. Such a war would probably be of very short duration, and the opportunity to recover from one's mistakes or to exploit the mistakes of the other side would be very limited. The side that is best prepared, including having the best doctrine, is most likely to prevail. Having an operating model with which to evaluate our doctrine is therefore a benefit to our Armed Forces.

One of the basic design concepts of the model is that all entities, whether they be units, terrain, or man-made objects, will have comparable units of measure. In formulating a plan for an attack of the opponent's rear area, a commander has to decide which targets to attack. Any reasonable algorithm for making this decision will demand that all targets be measured in comparable units. As the targets are likely to be a heterogeneous mix of entities, having a common unit of measure is imperative. A system to establish unit values in common metrics, called the Generalized Value System, has been designed and initially tested for use in ALARM [Ref. 3]. The measure of a unit's capability is called its POWER. This is measured in Standard Units of Power, or STAPOWS. The power of a unit in any two situations will likely not be the same, so a method of computing a situationally-inherent power has been formulated. There are many factors that go into this computation, such as the type of support available to a unit, the mission it is assigned, the mission that it performs best, and so forth. One of the traditional difficulties in determining the benefit of attacking support units in rear areas has been the inability to determine their contribution to combat units. The Generalized Value System includes the concept of derived power to overcome this problem. A support unit *derives* its power from the power of the combat units that it supports.

A second basic design feature concerns the ability to forecast events. In most current models, the only information available to a decisionmaker is the current status

of the forces engaged. The information about the future state of any given entity is noticeably absent. A conscious effort has been made in ALARM to establish mathematical relationships that predict the state of any entity at any point in time. This has made it possible to attempt the sort of decisionmaking envisioned in the AirLand Battle doctrine. A commander can begin to make plans for the future because he can forecast the status of every unit and can deal with situations that might result in a lost war.

C.   STATEMENT OF THE PROBLEM

One of the difficult problems the commander must solve in implementing AirLand Battle doctrine is determining which targets should be attacked because their destruction provides the maximum benefit to his force. A further dimension to the problem is deciding, once the targets are picked, which of his assets should be used to prosecute the attack. In making this decision, he should consider units that can make a successful attack, simultaneously suffering the least damage themselves. Still another dimension of the problem is deciding when is the most opportune time to make the attack. Time becomes a complicated problem because it must be considered both with regard to the enemy forces and also to the friendly forces.

The goal of this thesis is to develop an algorithm that can determine what asset-target assignments provide the maximum benefit to the friendly force. The algorithm is designed specifically to consider artillery assets, but it has the potential to be used with every type of asset that must be allocated to missions or targets in a battle. The significant factors in the decisionmaking process are considered and are converted to mathematical expressions for the algorithm. Rules for assigning standard Field Artillery missions have been developed that use the output of an optimization process to determine mission assignments. A number of optimization techniques are considered. The continuous nature of the equations that describe a unit's power over time, which are developed using the Generalized Value System and Lanchester Attrition Processes, and the reality of fighting an enemy free to select his own optimal strategy, led to the selection of the method of differential games as the optimizing tool. This method is imbedded in a rule-based decision algorithm that utilizes user-selected thresholds to select missions for artillery units. The rules reflect the commander's goals of attaining a specified decrease in the enemy's power in a limited time window, while minimizing the amount of power expended by his own units.

10

D.   OUTLINE FOR THE THESIS

The development of the algorithm to allocate assets to missions and targets will begin by considering the methods used in current models to make asset allocations. General optimization processes are outlined and their applicability to the Generalized Value System is analyzed, and the method of differential games is selected as the optimizing process. The process of making allocation decisions in actual practice and what causes the process to be initiated is described. This description of the decisionmaking process is converted to actual algorithm steps with the differential game imbedded as the optimizer. The method of differential games is explained and the equations used in this application are derived. An example of an artillery allocation problem is outlined and the algorithm is used to solve it. Analysis of the results suggests areas for further research.

II. ALLOCATION DECISIONS IN MODELS

A.   ALLOCATION IN CURRENT MODELS

The purpose of an allocation algorithm is to provide assistance to a decisionmaker in assigning his assets to missions or targets. In some cases, there is very little information available to the decisionmaker, so the number of choices reduces to only a few, and the decisionmaker is able to discern the optimum without the aid of an algorithm. In other cases, the choices themselves may be so limited that the solution is obvious. An example of this is when there is only one asset to be allocated. The harder cases, with several assets and numerous targets, as well as several factors that must be considered in each case, are the ones that demand the help of an algorithm, and they will be investigated further in this thesis.

Previous models have generally approached the problem of allocating assets to targets by first establishing a set of prioritization rules for each asset type. For example, artillery units might have one set of rules to determine which targets should be attacked, while attack helicopter units have a different set. This method of determining asset-target allocations worked as long as the process started with one asset and multiple targets. The algorithm simply sorts the targets in order according to the rules and breaks ties with some additional rule. The method is not so clear when there are multiple assets and one target. Here it is not a case of sorting targets, but of sorting assets to find the one that is best according to the decisionmaker's utility. The rules for determining the optimal asset-target assignment that were used in the preceding case cannot be used, and a new set must be formulated.

A second shortcoming of previous models has been their inability to make future plans based on the forecasted future value of the targets. The strength and value of entities is often based on a 'snapshot' of the battle, meaning an estimate at a given instant of time. The best any model can do under this constraint is to give the current state of the entities. Unless there exists a means of extrapolating forward in time, the commander is forced to make his decisions based solely on this data. This is a departure from actual practice where the past states of an entity can be considered along with the current, and a projection into the future is made. An example of this can be seen by considering a bridge, usually a critical entity on a battlefield. Assume

that the bridge was made unusable by an air attack at time T_1. An engineer unit is dispatched to repair the bridge, and is in the process of doing so. A 'snapshot' of the battlefield would show the bridge being unusable and would not credit it with being partially repaired. Most models would not show that the bridge is gradually gaining value and strength as the repairs progress, until it becomes a fully functional entity. With the Generalized Value System and forecasting, the decision algorithm will be able to consider the bridge and its value as a continuous function over time.

B.   METHODS OF OBTAINING OPTIMAL ALLOCATIONS

There are several techniques that can be applied to provide an optimal solution to the asset-target problem. The most basic method is to use linear or nonlinear programming, depending on the formulation of the problem. One of the characteristics of these methods is that they are essentially static. The modeler cannot specify continuous time in his model. One way to get around this problem is to make time discrete and solve the linear or nonlinear program for each discrete time period. A further complication can arise if there are many strategies for the two sides to use. This can lead to the specification of so many strategies that the programming solver is overwhelmed.

A more sophisticated technique is optimal control theory. It has the advantage of treating time continuously. It is different from other techniques in that it only considers one side of the conflict to be a rational decisionmaker, while the opponent is considered to follow a set of predetermined courses of action. The opponent does not have the ability to alter his course of action during the game in response to the game situation.

Another sophisticated technique is known as the method of differential games. It also treats time continuously and has the advantage of allowing rational decisionmakers on both sides of the conflict. Each side has an objective it is to achieve, generally the opposite of the opponent's objective. This feature has appeal to the military planner, who should be basing his plans on the enemy's capabilities until he is certain of the enemy's intentions, which may not become apparent until it is too late to react.

Optimal control theory and differential games offer attractive features that apply to the allocation problem, and have been explored as tools to be used in solutions. They both can handle the dynamic nature of combat, specifically the equations of

Lanchester [Ref. 4:pp. 55-63] which will be used in ALARM. An extensive study of differential games and their applications to military problems, particularly allocation problems, has been conducted by James G. Taylor. [Ref. 5]

C. THE ALLOCATION DECISION PROCESS

1. Demand on the System

The underlying purpose of this algorithm is to assist a decisionmaker in allocating his assets to a set of targets. To create a useful algorithm, it is important that the context of the decision is understood. In actual practice in Army units, the allocation of assets as part of a plan is driven by the perceived state of the friendly and opposing forces. The details of the allocation are specified to create a 'win' for the friendly forces, and consequently a loss for the opposition.

There are at least two decisionmakers involved in allocating the assets to targets. The overall decisionmaker is the ground force commander. He is in charge of the total ground combat force and is primarily responsible for the conduct of the battle. The commander of the particular asset to be committed is the functional area decisionmaker, such as a division artillery commander or an attack helicopter company commander. He is given a mission or goal by the ground force commander and determines what the optimal solution using his asset would be. To differentiate between the two in the remainder of the paper, they will be referred to as the force commander and the asset commander, respectively.

The force commander perceives that on some parts of the battlefield, his forces have the advantage over the enemy, and on other portions the enemy has the advantage. In those situations where the enemy has the advantage and the force commander has uncommitted assets available, the force commander should consider the possible uses of the assets and how they can best benefit him. This is analogous in a way to repairing a dike. If the reservoir is full and the engineer knows the stresses on the dike, he can determine where to put the materials to strengthen the dike so that they serve his needs best. A demand for the materials exists. Similarly, the force commander perceives the demand on his uncommitted assets to strengthen the units that are in a conflict they will lose.

2. The Decisionmaker's Objective

The demand must be expressed quantitatively for a mathematical algorithm to assist in solving the problem. This raises the issue of assigning a number to the demand

that makes actual sense. Each commander can probably arrive at a way of doing this, but the method should be characterized by common sense and simplicity. Consider the phrase 'optimal allocation'. In actual warfare use, an 'optimal allocation' is generally the one that defeats the enemy with the smallest expenditure of resources. Defeating the enemy is also a vague term. How is defeat expressed in numbers? A way of looking at this is that a force commander, using the forecast of the power of his force and the enemy force, decides that an enemy unit will have more power than one of his units at some time in the future, as in Figure 2.1, and he wants to optimally allocate his uncommitted assets to prevent that, or if he doesn't have enough assets available, he wants to request more from his superior. The difference in power quantities at the specified future time between his unit and the enemy's unit represents the difference between losing the battle and preventing the enemy from achieving his goals. Simply, if he has enough uncommitted assets and he allocates them to attack the enemy's units, then the enemy will not have more power than his unit at the future time, and he can assure at least a draw at that point in the battle. Further, by allocating his assets in an optimal manner, he may have some uncommitted assets remaining in the future. These could be allocated to deal with the actual situation as it becomes clearer. For the purposes of an algorithm that is to be used to allocate artillery fires, the goal of the asset commander will be to cause a decrease in the power of the opposing force to a specified level.

3. Constraints

As with many optimization problems, there are constraints that must be considered. First, the power of the forces on both sides is constrained to be nonnegative. A unit can have zero power when it is destroyed, and it can have any reasonable amount of positive power otherwise.

Second, each friendly artillery unit will be constrained in the amount and types of ammunition available. Available ammunition includes the ammunition on hand in the asset unit and ammunition that is in transit to the unit and will arrive before it is needed. Indirect ammunition constraints will be imposed on the opposing forces. There are not any ammunition counters for the opposing force, but with the concept of derived power for supporting units it is possible to logistically constrain any unit without actually counting quantities of ammunition or fuel.

A final constraint is that firing units may only attack those targets that are within range of the weapon system. This constraint exists for obvious reasons.
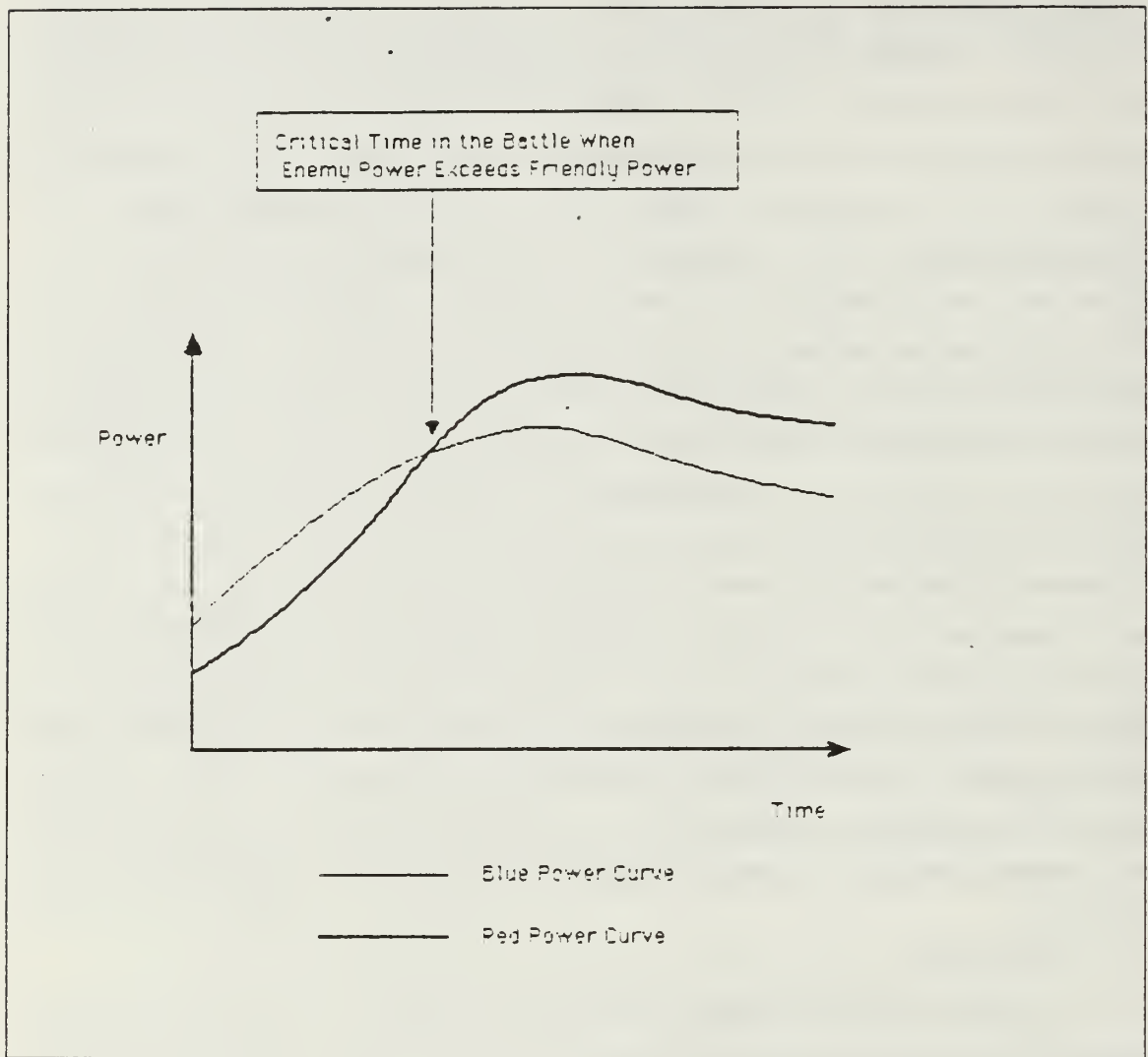
Figure 2.1 Graph of Blue and Red Power Curves.

D. INCORPORATING REALITY

The factors considered by an algorithm determine how closely it models reality. This can be a two-edged sword. An algorithm that tries to consider every factor, not just the important ones, is not responsive. It has as little value as one that does not consider enough factors.

1. Decision Parameters

One factor that is essential to the solution of the problem is time. In almost any real military problem, time is a scarce resource. In this problem, the force commander recognizes that at a specified time in the future, one or more of his units

16

will be 'overpowered' by the enemy. He desires to take action between now and that future time to prevent that outcome. The asset commander must therefore either decrease the enemy's power in the timeframe imposed or report that he cannot, in which case it is envisioned that the force commander would look for another asset to perform the mission or request assistance from his superior. Another alternative for the force commander is to combine the attacks of two or more uncommitted assets to accomplish the mission. This alternative is more indicative of how this problem is addressed in reality, particularly in AirLand Battle doctrine, where simultaneous or sequential attacks by different types of forces that have complementary attributes is considered to be more powerful than an attack by only one force.

A second factor to be considered is the enemy's power and how it is changing over time. With the Generalized Value System, it is possible to model the military intelligence section estimates of the status of enemy units and forecast their power in the future. Implied in that process is a judgement about the way the enemy's power is changing over time. It may be decreasing as he consumes supplies or increasing as he approaches the time and place where he begins to accomplish his mission. Also, by making similar judgements about the state of the enemy's logistics, power changes due to resupply or generally increased support may be indicated. Knowing this would give the asset commander the option of attacking a logistics unit, a target that may be easier to destroy and much less likely to return fire. This would be an indirect means of reducing an enemy unit's power. This is all important information for a decisionmaker, who should be looking for the time and place that gives the greatest payoff for using his asset's power. An optimal allocation of power can be found by finding those times and places, and attacking them in sequence.

Every attack carries an implied risk to the attacker. For an artillery unit about to fire for the first time on a target, part of the payoff for making the attack is the negative return of disclosing the artillery unit's position and creating the possibility that the enemy will detect it and return fire, with the resultant decrease in the power of the artillery unit. This possibility increases with time (i.e., as the artillery unit fires more rounds at the enemy unit, the more opportunity there is for the enemy to detect the exact location of the artillery unit). If the enemy returns fire accurately, the attacking unit will inevitably sustain losses of equipment and personnel. These losses will be called 'permanent' losses, and of course there will be the complementary 'temporary' losses.

17

Temporary losses will include those that can be replaced in a short amount of time. The most common example of a temporary loss is the expenditure of ammunition. This is a loss that is anticipated and replacement ammunition is pushed forward by the logistics system from the first day of conflict. Units are expected to expend ammunition and fuel. Permanent losses, on the other hand, cannot be replaced as readily. Their occurrence may be anticipated, as casualties certainly are, but replacements are generally 'pulled' through the logistics system. Damaged or destroyed vehicles and other forms of equipment are good examples of 'permanent' losses.

Both permanent and temporary losses are important factors in the allocation decision because they will occur, and one of the stipulations on the asset commander is that he minimize the asset power used. There is often a strong relationship between the amount of time a unit is firing, thereby exposing its position, and the amount of damage it receives from counterfire. Minimizing firing time is a way to avoid a large amount of power lost due to enemy fire. One way to minimize firing time is to fire the ammunition that gives the maximum attrition of the enemy power per round fired. Temporary and permanent losses represent power losses to the asset commander, so he is very concerned about them. They are separated, though, because temporary power losses may be regained in time to execute other missions with some certainty. Since they are planned for, the military planner can expect replacements in a short time. The same cannot be said for permanent replacements.

Two real measurements also influence the asset decisionmaker. First, he can only attack targets that are within range of the assets he controls. Second, as previously discussed, he may only fire the types and amounts of ammunition that are in the unit's possession or are in a resupply convoy that is available to the unit before the ammunition is to be fired.

Finally, the asset decisionmaker must consider the need to attack targets that have an overriding priority. It is fairly common practice to establish a set of targets whose destruction is of benefit to the entire force, and therefore these targets are accorded a very high priority. An example of this might be a nuclear-capable missile battery or a radio-jammer. The nuclear-capable missile battery may only represent a fraction of the power of a tank regiment, but its potential for inflicting severe damage in a very short period of time makes it a target of immense importance.

2. Artillery Missions

Without going into a lengthy discussion of the Field Artillery and the way it is tactically employed, a short explanation of the subject is necessary for understanding the algorithm. This discussion will cover the missions artillery units are given, the general rules used in determining what missions are assigned, and a brief example of how they will be modelled in the algorithm.

There are four standard missions that may be given to a field artillery unit. They are Direct Support, Reinforcing, General Support-Reinforcing, and General Support. The actual differences between each of the misssions can be found in U.S. Army Field Manual 6-20, *Fire Support in Combined Arms Operations.* Direct Support is the relationship that usually exists between an artillery battalion or brigade and a maneuver brigade. It implies that the first and primary responsibility of the artillery unit is to support the maneuver brigade. Reinforcing is a mission that can be given to a field artillery unit when that unit is to provide primary support to another field artillery unit, which is itself in direct support of a maneuver brigade. General Support is the mission given when a unit is to provide support to the entire organization, not just a portion. This commonly occurs at the level of Division or Corps. An artillery unit might be given the mission of General Support to the Division, meaning it provides support to every brigade, not just a specific one. Finally, General Support-Reinforcing is a mix of the two preceding missions. A unit with this mission provides primarily general support to the entire organization, but secondarily provides reinforcing fires to a specific field artillery unit in direct support to a maneuver brigade. [Ref. 6:p. C-7]

A set of rules or guidelines exist in the Field Manual cited that are used to determine mission assignments. As a rule, missions are assigned to artillery battalions or brigades, and the subordinate units have the same mission as the parent unit unless otherwise specified. The first mission assignment rule is to maintain the maximum feasible central control. Artillery is most effective when it attacks in mass, and centralizing control facilitates such attacks. The second mission assignment rule is that a field artillery unit will be assigned in direct support to each committed maneuver brigade. If a brigade is not committed, it will not have any direct support artillery until it is committed. The third rule is to weight the main avenue of attack (in the offense) or the most threatened sector (in the defense). This is normally done by assigning a mission of Reinforcing or General Support-Reinforcing to one or more units. The fourth rule is to assign missions to facilitate future operations. The fifth rule is to keep

some artillery available to the force commander to influence the battle. This is generally accomplished by assigning one or more units a General Support or General Support-Reinforcing mission. The final rule is to keep no artillery units in reserve. [Ref. 6:pp. C-10-12]

The way the algorithm will assist in mission assignments is by solving a differential game to determine the optimal allocation of assets to targets, then assigning missions based on thresholds. These thresholds might be time-specific or power-specific. For example, the algorithm first assigns a Field Artillery battalion in direct support of each maneuver brigade. Then it solves the game and returns the solution that specifies what artillery units fired what targets over the time span of interest. If an uncommitted unit fired for more than a specific percentage of time, say 50%, at targets in the First Brigade sector, then it would be assigned the mission of Reinforcing the direct support battalion assigned to the First Brigade. Or if a unit fired at targets in each sector in basically equal amounts, it would be assigned the mission of General Support. The specific thresholds and percentages used in the algorithm should be provided by the user.

With an understanding of the techniques available for solving an allocation process and the framework and factors of the decision process, the next step is to develop the algorithm and explain the tools used in it.

III. ALGORITHM DEVELOPMENT

The development of an algorithm to resolve the optimal allocation problem requires that the decisionmaking objectives, constraints, and factors be translated into algorithm steps in a simple yet complete form. It also requires that a technique of solving allocation problems be selected and implemented. The technique that will be used is the method of differential games, as described in Chapter 2, Section B.

A. DEVELOPMENT OF ALGORITHM STEPS

The need for this algorithm arises when the force commander determines that a demand exists for the use of his uncommitted assets. Because ALARM uses standard units of power (STAPOWS) as a measure of a unit's strength, this demand should assume the form of "decrease the enemy force by ΔY STAPOWS". Since the force commander projects the enemy's power and his own force's power forward in time to determine the amount of power decrease required, he will specify a time by which the decrease in power must be accomplished. Given the current state of the various forces, the power decrease must be completed by a specified future time in order for the force commander's objective to be satisfied.

1. Inputs to the Algorithm

The first step in the algorithm is to acquire the information needed to make the allocation decision. The asset commander who makes the decision first receives the mission from the force commander.

"Decrease the enemy's force at time, t_f (the future time) by ΔY STAPOWS." This statement contains the first two inputs to the algorithm. One is the required power decrease in the enemy force, and the second is the time by when the power decrease is necessary.

A further set of inputs is the power level of each of the units involved in the allocation decision as either assets or targets. These power levels are available through the Generalized Value System for both the current time and the end of the timeframe, t_f, under consideration.

There are several other items of information that are needed to solve the algorithm. Their uses will be explained in greater detail as the algorithm is developed. They include:

21

- the current locations of the units involved in the allocation as either assets or targets, and their direction and rate of movement if they are moving;

- the maximum range of each unit's weapons;

- the amounts and types of ammunition in each unit's possession;

- attrition rates for each ammunition-target and asset-target combination;

- the fraction of total power that is represented by the ammunition on hand in each friendly unit;

- the number of firing systems available in each friendly unit;

- the rate of fire for the weapon system in each friendly unit.

2. Feasibility Checks

The first step in allocating assets to targets is to eliminate from consideration those targets that are beyond the maximum range of a unit's weapons. The information needed for this step is the current positions of every asset and target, the direction and rate of movement if a unit is moving, and the maximum range of each unit's weapons. If a target is beyond the range of a unit's weapons, the attrition coefficient for that asset-target combination is set to zero. This will result in the pair being nonoptimal in the differential game. This step can be repeated after each time period.

A second feasibility check applies to ammunition selection for firing. The attrition coefficient, α_{ij}, which is the attrition rate of enemy target j when fired on by friendly unit i, is linked to the type of ammunition fired by unit i. For example, if unit 1 fires a high explosive round against an enemy tank unit 2, α_{12} might be 0.02. If a precision guided round were fired, α_{12} might be 0.1.

Ammunition selection for each Blue artillery unit will be constrained to the ammunition that is actually in the unit's possession in the algorithm. A full model may consider not only the ammunition on hand, but also the ammunition that is being sent to the unit. Because the algorithm is being demonstrated in a limited scenario, the full logistical package necessary to represent the resupply of ammunition is not yet available. In the future, ammunition resupply to the Blue forces will be considered.

The algorithm will calculate the amount of ammunition necessary for all weapon systems to fire at a specified rate of fire in the next time period. This will be checked against the amount on hand for each ammunition type, and those that do not exist in sufficient quantities will not be considered for firing. For example, artillery unit X_2 has twenty-four (24) howitzers available and an individual howitzer fires at a rate of one-half (.5) of a round per minute. The formula

Ammo required = (rate of fire per system)*(number of systems)*time period

is applied with the result that twelve (12) rounds are needed in unit X_2 for every minute of firing. The ammunition quantities in X_2 are checked against this amount required, and any type that is not on hand in the amount required is eliminated from consideration by setting α_{2j} to zero.

3. Determining Attrition Coefficients

There is no mechanism in the algorithm for counting the quantity of ammunition expended by the enemy units. The power of the ammunition they use is accounted for by the derived power of their logistics units. The attrition coefficients β_{ji} are based solely on the firer-target combinations. The type of ammunition fired is not considered.

For Blue units the rate at which Red targets are attrited is linked to the ammunition fired against the target. This is a natural linkage, since artillery units damage or destroy enemy forces by delivering indirect fires to the target. Different types of ammunition have differing effectiveness against the same target, as has already been shown, and tables are used in manual or automated ammunition selection to find the best combination. The general rule of thumb is to select the ammunition that has the highest effectiveness. This is closely related to selecting the ammunition that has the highest attrition rate.

The method the algorithm uses to find the attrition coefficient for the Blue unit is to select, from the ammunition types that are on hand in the required quantity, the ammunition type that has the maximum attrition rate. There are two justifications for this selection rule. First, depending on the firer's motivation, this ammunition type will give the maximum attrition over a fixed time interval, or it will require the shortest firing time to attain a specified amount of attrition. The Blue asset commander, for reasons explained in Chapter 2, Section D, Subsection 1, will endeavor to minimize firing time, so he wants to select the ammunition that provides the maximum attrition rate. Secondly, the method used in the algorithm for determining which asset-target allocations are optimal requires that the maximum value of a product whose terms include α_{ij} be found. To insure that this occurs, the value of α_{ij} should be a maximum.

4. Determining Power Loss Due to Ammunition Expenditure

The next step in the algorithm is to determine the optimal asset-target allocation. The method for doing this is part of the explanation of the technique of differential games, explained later in this chapter. For now, assume that the optimal asset-target combinations have been specified.

23

Once the best asset-target combinations have been selected, the ammunition that each asset will fire is determined. In the preceding steps, the best ammunition each asset should fire on every possible target was determined. Now that the actual target is known, the calculation for power lost due to ammunition expenditure in a time period is given by:

Power Loss = (power of one round of type k)*(number of rounds fired in time period)

One of the inputs to the algorithm is the fraction of total power of each Blue unit that the ammunition on hand represents. The power that the total amount of each type k ammunition represents is proportional to the ratio of the type k ammunition quantity to the total ammunition quantity, multiplied by a constant that represents the value of the type k ammunition relative to all other types.

Total power of ammunition type k = (Total ammunition power)*(Quantity of type k ammunition)*(Relative Weight)/(Total ammunition quantity)

The unit of measure of total power of type k ammunition is STAPOWs. The power in each round is found by dividing the power of the type k ammunition by the number of rounds of type k ammunition. This will result in an equal division of power to each round in the same ammunition type.

After this step in the algorithm, book-keeping steps are taken to update the total ammunition on hand, the total amount on hand in each type, and the power of the ammunition remaining. This update is done to ensure that the planned solution remains within the feasible limits for ammunition.

5. Determining Optimal Allocations

All of the information required to formulate the differential game is now available. The procedures for solving the game will be presented in detail in Section B. The game will specify as output what asset-target allocations are optimal. Since these allocations are expected to change as time progresses, the output will specify when the changes occur and what the new combinations are after the change.

6. Mission Assignments

The final step in the algorithm is to assign missions to uncommitted units. In solving the problem for an artillery decisionmaker, the algorithm compares the time or the power, at the user's direction, spent by each asset engaging targets in the threatened sector with the threshold parameters. It assigns missions to the unit when it exceeds the mission threshold.

In the event that the power available in the uncommitted assets is not sufficient to attain the goal of Red power decrease specified, the algorithm returns a result that states this fact. Figure 3.1 is a concise representation of the algorithm.

B.   USING A DIFFERENTIAL GAME TO FIND OPTIMAL ALLOCATIONS

The core of the algorithm is the differential game that is used to determine the optimal allocations. There are two features of ALARM that lead to the selection of this technique. First, the power functions of the Generalized Value System and the equations of dynamic combat developed by Lanchester lead to the consideration of an entity's power as a continuous function over time. It seems logical that the method of allocating assets to targets should take advantage of the continuous nature of these functions. Linear and nonlinear programming do not. Secondly, the nature of warfare is such that both the Blue and Red commanders are striving to attain their objectives and are not locked into a predetermined strategy. They can both make decisions about allocating their resources in response to their opponent. The theory of optimal control only allows one of the decisionmakers to react to the opponent. A differential game incorporates these desirable features.

1. Power Equations

Every entity involved in the allocation decision is represented in the algorithm. There are a set of Blue units, represented by X_1, X_2,..., X_m, and a set of Red units, represented by Y_1, Y_2,..., Y_n. The Blue units are the assets to be allocated, and the Red units are the potential targets. Other entities may be represented, such as bridges, airfields, or cities. They will be included as assets if they contribute to Blue power, or targets if they contribute to Red power. The variable X_i (or Y_j) represents both the identification of the entity and the power it possesses.

The power of every entity can be expressed as a function of time according to the equations developed in the Generalized Value System. In this system, there are several types of power, the definitions for which are in Appendix 2. The power used in the algorithm is the Situational Inherent Power, defined to be ". . . the prediction, at time t_p, of the inherent power that an entity X_1 will have at time t, given the state of the entity at t_p, SX_1 (t_p); t_p < t." [Ref. 3]. The present time, or the time the prediction is made, is t_p. The time that the prediction applies to is t. There is a third time that is important, because it is the time when the unit reaches the maximum power it can have. This is the time, denoted t_a, when a unit at full strength is in a
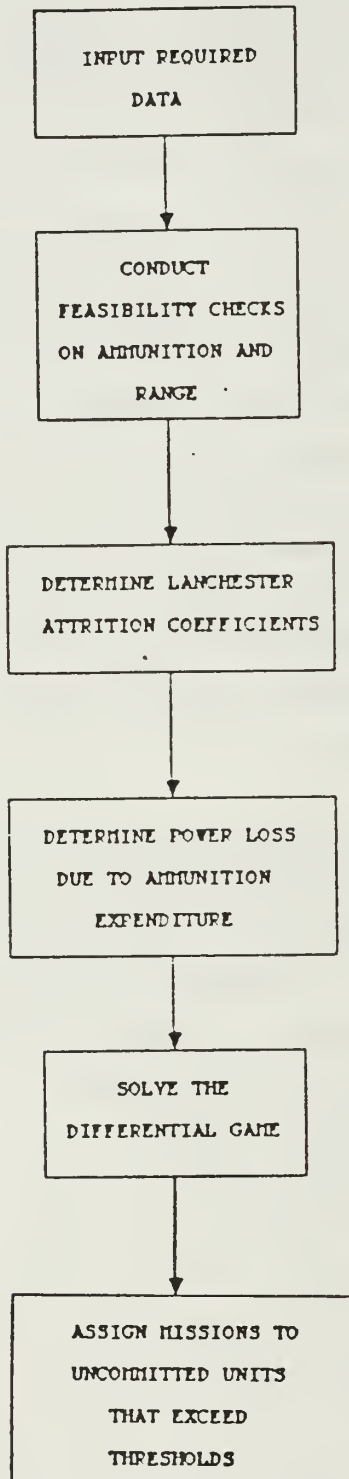
Figure 3.1   Allocation Algorithm.

position to start accomplishing its assigned mission. The equation for the Situational Inherent Power is:

$$SIP(X_i(t)|SX_i(t_p)) = PABIP(X_i(t)|SX_i(t_p))*exp(-D_i(t_a-t)), \quad t < t_a \qquad \text{(eqn 3.1)}$$

If $t \geq t_a$, then

$$SIP(X_i(t)|SX_i(t_p)) = PABIP(X_i(t)|SX_i(t_p)) \qquad \text{(eqn 3.2)}$$

(The mnemonic PABIP stands for Predicted Adjusted Basic Inherent Power and is defined in Appendix 2.) The term D_i represents the rate at which the unit is attaining readiness as it approaches the time and place when it attains maximum power. It is somewhat analogous to a discounting factor [Ref. 3:p. 38]. Using equation 3.1, the power of an entity as it approaches the time and place where its mission begins can be determined.

The other process that determines the way an entity's power changes over time is attrition due to combat. The algorithm uses the equations of attrition developed by F. W. Lanchester [Ref. 4:pp. 52-60]. Since it is solving the problem of allocating artillery fires, the Linear Law formula is used in the algorithm. This implies that each side fires into an area, instead of employing aimed fire. This is acceptable unless the artillery is firing ammunition that receives guidance to a specific target by some means, such as Copperhead or the proposed SADARM projectiles. In that case, a Square Law formula seems more appropriate. A likely compromise on this in a future application could be the Helmbold equations [Ref. 4:p. 175]. At this time, the algorithm does not include provisions for such ammunition.

The Lanchester Linear Law equation for a Blue entity, X_i, opposed by Red entities, Y_j, is:

$$dX_i/dt = -\sum_{j=1}^{n} (\beta_{ji}*X_i*Y_j) \qquad \text{(eqn 3.3)}$$

A similar equation can be developed for every entity in both forces.

The equation for the total change in power of an entity is a combination of Equations 3.1 and 3.3. The solution mechanism requires a differential equation to

express the power of every entity, so the Situational Inherent Power equation is differentiated with respect to time, with the result:

$$d/dt\{SIP(X_i(t)|SX_i(t_p))\} = d/dt\{PABIP(X_i(t)|SX_i(t_p))^*exp((-D_i)^*(t_a-t))\} \quad \text{(eqn 3.4)}$$

$$+ PABIP(X_i(t)|SX_i(t_p))^*d/dt(exp(-D_i^*(t_a-t)))$$

The term $PABIP(X_i(t)|\ SX_i(t_p))$ will be considered a constant, so this reduces to:

$$d/dt\{SIP(X_i(t)|SX_i(t_p))\} = \{PABIP(X_i(t)|SX_i(t_p))^*D_i\}^*exp(-D_i^*(t_a-t)) \quad \text{(eqn 3.5)}$$

This can be combined with the Lanchester equations 3.3 to obtain the following expression for the change in power of entities:

$$dX_i/dt = -\sum_{j=1}^{n}(\beta_{ji}^*X_i^*Y_j) + PABIP(X_i(t)|SX_i(t_p)) \qquad \text{(eqn 3.6)}$$
$$^*[exp(-D_i^*(t_a-t))]^*D_i$$

and for Yj:

$$dY_j/dt = -\sum_{i=1}^{m}(\alpha_{ij}^*Y_j^*X_i) + PABIP(Y_j(t)|SY_j(t_p)) \qquad \text{(eqn 3.7)}$$
$$^*[exp(-D_j^*(t_a-t))]^*D_j$$

As was mentioned in Section 1 of this chapter, the algorithm also determines the power loss due to the expenditure of ammunition by each Blue unit. This quantity is determined by computing the power represented by a single round of ammunition and multiplying that quantity by the number of rounds fired in a time period. This product is calculated for each ammunition type fired and the products are summed, resulting in the power loss due to ammunition expenditure in the time period:

$$\text{Power Loss} = -\sum_{l=1}^{k} a_{il}^*n_l \qquad \text{(eqn 3.8)}$$

where a_{il} is the power of one round of type l in unit i, and n_l is the number of rounds fired in the time period.

This expression for power lost is added to equation 3.6 for power change for the Blue units:

$$dX_i/dt = -\sum_{j=1}^{n} \beta_{ji}*X_i*Y_j + PABIP(X_i(t)|SX_i(t_p))$$

$$*(exp(-D_i*(t_a-t))*D_i) - \sum_{l=1}^{k} a_{il}*n_l \qquad \text{(eqn 3.9)}$$

Equations 3.9 and 3.7 for dX_i/dt and dY_j/dt represent the change in the state variables X_i and Y_j as the battle progresses in time. The purpose of the algorithm is to determine when Blue units should fire at Red targets, and what targets should be engaged, so the required attrition occurs while minimizing the power expended by the Blue units. As equations 3.9 and 3.7 (known as Kinematic equations) now stand, there is no means for Blue to selectively fire at Red, or Red at Blue.

The means for doing this is to introduce control variables. Blue will indicate selection by the value of the control variable φ and Red will indicate selection by the value of the control variable ψ. The value of φ_{ij} will determine when Red unit j is selected as a target for Blue unit i and the opposite meaning holds for ψ_{ji}. In practice the subordinates of each Blue or Red unit will have the same mission or target as the parent unit. If the Blue artillery battalion is firing on a target, the entire battalion will be firing on it, not a fraction of it. The only exception to this is when one of the Blue batteries is moving. The algorithm will ignore this exception, since it is involved in planning and not actual execution. The possible values of φ_{ij} and ψ_{ji} will be zero or one.

The final form of the Kinematic equations is then:

$$dX_i/dt = -\sum_{j=1}^{n} \beta_{ji}*\psi_{ji}*X_i*Y_j + [PABIP(X_i(t)|SX_i(t_p))]$$

$$*[(exp(-D_i*(t_a-t)))*D_i] - \sum_{l=1}^{k} a_{il}*n_l \qquad \text{(eqn 3.10)}$$

29

$$dY_j/dt = -\sum_{i=1}^{m} \alpha_{ij} {}^* \varphi_{ij} {}^* Y_j {}^* X_i + [PABIP(Y_j(t)|SY_j(t_p)]$$

$$^*((\exp(-D_j {}^*(t_a-t))){}^*D_j) \qquad \text{(eqn 3.11)}$$

2. The Terminal Condition

If the differential game starts at the current time, which is t_0, it will progress by means of the Kinematic equations until it reaches the desired terminal conditions. In this algorithm, the terminal conditions are bounded by the constraints that the Blue and Red units have nonnegative power quantities and t must be greater than t_0. With the Blue goal of decreasing the power of the Red forces by a specific amount, an additional terminal condition is that Red's final power must be less than or equal to the maximum allowable amount. Since there is a time limit on achieving the attrition, the condition that the game must end by a specified time also exists. Figure 3.2 is a general depiction of the surface of the game in two dimensions, showing the Red force power decrease and the time of the game.

Figure 3.2 depicts the power of the Red force from t_0 to t_f. $Y(t_f)$ is the maximum allowable power of the Red force at t_f. In general, there are some points $Y(t_0)$ from which it is possible to decrease Red's power to $Y(t_f)$ at or before t_f, and there are some points $Y(t_0)$ for which it is not possible to attain $Y(t_f)$ at or before t_f. For the Blue asset commander, this equates to a difference between attaining the required power decrease in the specified time or not attaining it.

3. The Payoff of the Game

Differential games are solved recursively, so the terminal conditions will become the initial conditions for the algorithm. The constraint on time is removed, and the payoff to each side will be the time required for Blue to cause the desired attrition to Red. The Blue commander wants to minimize the time needed to reduce Red's power to the necessary level. Red, on the other hand, wants to maximize the time required to attain the power decrease. The equation for the payoff is then:

$$\text{Payoff} = \int_0^T dt \qquad \text{(eqn 3.12)}$$
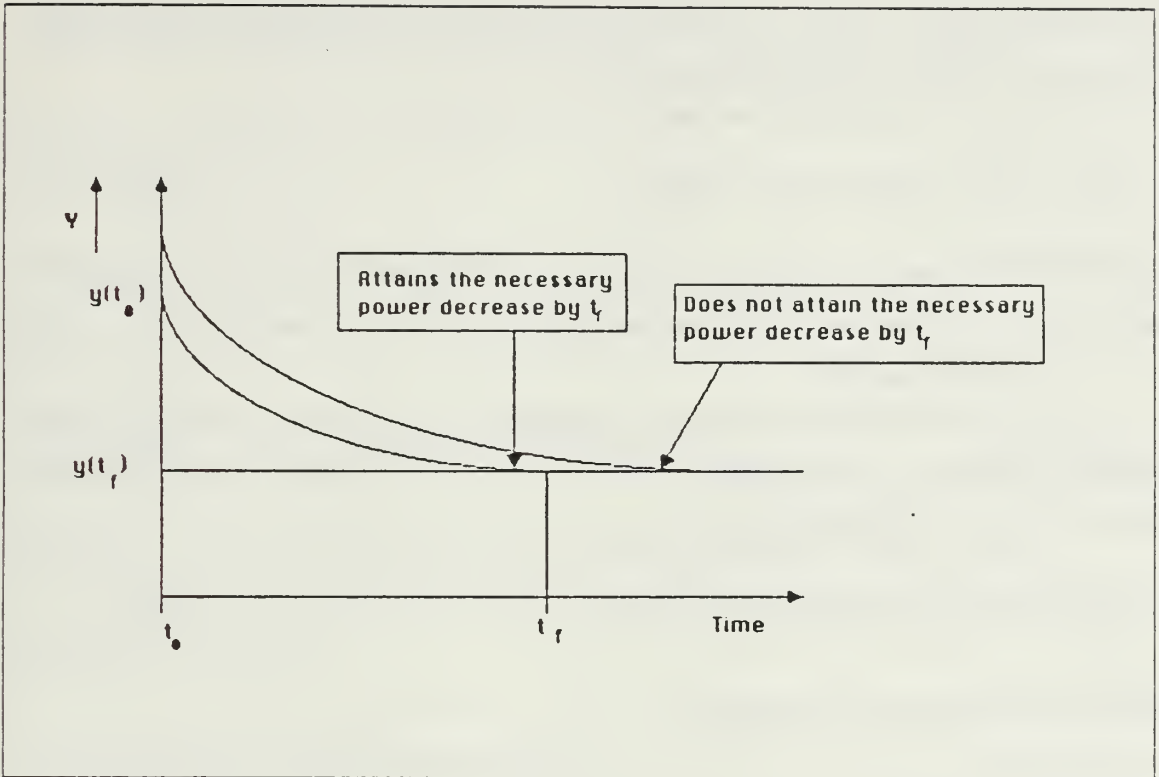
Figure 3.2   Terminal Surface.

4. Strategy and Value

In game theory the term 'strategy' means the decision that the player makes at each point in the game about how he will play [Ref. 7:p. 36]. For this problem, the decision to be made is whether to attack each of the possible targets. Blue indicates his strategy by setting φ to 1 if he will attack, or to 0 otherwise. Red makes the same choices on ψ. At the end of the battle, the strategy for the whole battle will be the set of φ and ψ values chosen. It is expected that φ_{ij} and ψ_{ji} will change during the course of the battle. The restriction is imposed that a firing unit may only fire on one target at a time. After the initial targets are selected, the optimal solution may include changing targets to get a better payoff. The time that the shift occurs and the shift itself are important to the asset commander. The entire set of allocations and the times that the allocations change comprise the strategy for each side over the whole battle.

The value of the game, or battle, occurs when Blue and Red both achieve the payoff they desire. The value for Blue is the minimum time to attain the attrition, regardless of Red's attempts to delay it. For Red it is the maximum time to attain the

31

attrition, regardless of Blue's attempts to hasten it. For each entity in the game, a value exists, and is denoted $V(X_i)$ or $V(Y_j)$. Because the payoff is dependent on the strategy each player uses, the expression for the Value is :

$$V(X_i) = V(Y_j) = \min_\varphi \max_\psi (\text{Payoff}) \qquad \text{(eqn 3.13)}$$

[Ref. 7:p. 36]. Hereafter, references to the Value of the game will use an uppercase V, and references to the value of all other quantities will use a lowercase v.

5. **The Main Equation**

Let all of the state variables be represented by the vector X. With the state variables known at t_0, it is possible to advance in time by Δt and determine the new value of the state variables, given by:

$$X(t_0 + \Delta t) = X(t_0) + (dX/dt)^* \Delta t = X_0 + \Delta X \qquad \text{(eqn 3.14)}$$

The Value of the game at this point is:

$$V(X(\Delta t)) = \int_{t_0}^{t_0 + \Delta t} dt = \Delta t. \qquad \text{(eqn 3.15)}$$

The game begins again with the new values of the state variables, and with both players using their optimal strategy. At the end of the game the total payoff will be:

$$V(X) = \Delta t + V(X_0 + \Delta X) \qquad \text{(eqn 3.16)}$$

and it can be shown that :

$$V(X_0 + \Delta X) = V(X_0) + \sum_i \partial V(X)/\partial X_i {}^* \Delta X_i \qquad \text{(eqn 3.17)}$$

$$= V(X_0) + \sum_i (\partial V(X)/\partial X_i)^*(dX_i/dt)^* \Delta t.$$

If the players use the optimal control variables in the first Δt of the game, then the total payoff of the game would be:

$$V(X) = \Delta t + V(X_0) + \sum_i (\partial\, V(X)/\partial\, X_i)^*(dX_i/dt)^*\Delta t \qquad \text{(eqn 3.18)}$$

Both sides of Equation 3.18 are divided by Δt and as the size of Δt approaches 0, Equation 3.18 reduces to:

$$0 = 1 + \sum_i (\partial\, V(X)/\partial\, X_i)^*(dX_i/dt) \qquad \text{(eqn 3.19)}$$

which is equivalent to:

$$\min_\varphi \max_\psi \{1 + \sum_i (\partial\, V(X)/\partial\, X_i)^*(dX_i/dt)\} = 0. \qquad \text{(eqn 3.20)}$$

Equations 3.19 and 3.20 are known as the Main Equation [Ref. 8:pp. 101-102].

The Main Equation of the algorithm, with $X = \{X_1, X_2, ..., X_m, Y_1, Y_2, ..., Y_n\}$ can be rewritten as:

$$\min_\varphi \max_\psi \{-\sum_{i=1}^m V_i[\sum_{j=1}^n \beta_{ji}^*\psi_{ji}^*X_i^*Y_j \qquad \text{(eqn 3.21)}$$

$$+ (B_i^*\exp(-D_i^*(t_a-t)))^*D_i + C_i] - \sum_{j=1}^n W_j[\sum_{i=1}^m \alpha_{ij}^*\varphi_{ij}^*Y_j$$

$$^*X_i + (R_j^*\exp(-D_j^*(t_a-t)))/D_j]\} = -1$$

where:

$$V_i = \partial\, V(X)/\partial\, X_i \qquad \text{(eqn 3.22)}$$

$$W_j = \partial\, V(X)/\partial\, Y_j \qquad \text{(eqn 3.23)}$$

$$B_i = PABIP(X_i(t)|SX_i(t_p)) \qquad \text{(eqn 3.24)}$$

33

$$R_j = PABIP(Y_j(t) | SY_j(t_p))$$ (eqn 3.25)

$$C_i = \sum_{l=1}^{k} a_{il}{}^* n_l$$ (eqn 3.26)

and the expressions for dX_i/dt and dY_j/dt have been substituted. This can be modified to:

$$\max_\psi \{ -\sum_i V_i{}^* [\sum_j (\beta_{ji}{}^* \psi_{ji}{}^* X_i{}^* Y_j) + B_i$$ (eqn 3.27)

$$^*((\exp(-D_i{}^*(t_a-t)))^* D_i) + C_i] \} + \min_\varphi \{ -\sum_j W_j{}^* [\sum_i (\alpha_{ij}{}^* \varphi_{ij}{}^* Y_j{}^* X_i)$$

$$+ R_j{}^*((\exp(-D_j{}^*(t_a-t)))^* D_j)] \} = -1$$

or, rearranging to group terms with the control variables present and multiplying by -1, an equivalent form is:

$$\sum_i V_i{}^* [B_i{}^*((\exp(-D_i{}^*(t_a-t)))^* D_i) + C_i] + \min_\psi (\sum_j Y_j{}^* \sum_i V_i$$ (eqn 3.28)

$$^* \psi_{ji}{}^* \beta_{ji}{}^* X_i) + \sum_j W_j{}^* [R_j{}^*(\exp(-D_j{}^*(t_a-t)))^* D_j] +$$

$$\max_\varphi (\sum_i X_i{}^* \sum_j (W_j{}^* \varphi_{ij}{}^* \alpha_{ij}{}^* Y_j)) = 1$$

Considering only the final term of Equation 3.28, the way to obtain the maximum value for that quantity subject to the constraint that $\varphi_{ij} = 0$ or 1 is to find j such that $W_j{}^* \alpha_{ij}{}^* Y_j$ is a maximum, and make $\varphi_{ij} = 1$ for that j. The same idea holds for the second term in the left-hand side. For each j, find the i value for which $V_i{}^* \beta_{ji}{}^* X_i$ is the minimum, and set $\psi_{ji} = 1$ for that i.

The algorithm thus has a rule for setting the control variables for both players. The values of α_{ij} and β_{ji} are known inputs, and the values for each X_i and Y_j can be found, at every point in time, by integrating the expressions for dX_i/dt and dY_j/dt. That leaves the issue of determining the values of V_i and W_j.

6. The Path Equations

Recall the Main Equation:

$$1 + \sum_i (\partial V(X)/\partial X_i)*(dX_i/dt) = 0 \qquad \text{(eqn 3.29)}$$

If the left-hand side of Equation 3.29 is differentiated with respect to X_j, $j \neq i$, the result is the sum:

$$\partial/\partial X_j (\sum_i (\partial V(X)/\partial X_i)*dX_i/dt) = 0 \qquad \text{(eqn 3.30)}$$

Applying the Chain Rule, this becomes:

$$\sum_i (\partial^2 V(X)/\partial X_i * \partial X_j)*(dX_i/dt) \qquad \text{(eqn 3.31)}$$

$$+ \sum_i (\partial V(X)/\partial X_i)*(\partial/\partial X_j (dX_i/dt))$$

$$+ \sum_k \partial/\partial\Phi(1 + \sum_i (\partial V(X)/\partial X_i)$$

$$*(dX_i/dt))*\partial\Phi/\partial X_j + \sum_l \partial/\partial\Psi(1 + \sum_i (\partial V(X)/\partial X_i)*(dX_i/dt))*\partial\Psi/\partial X_j = 0$$

where Φ and Ψ denote the vectors of control variables $(\varphi_1, \varphi_2, ..., \varphi_m)$ and $(\psi_1, \psi_2, ..., \psi_n)$. The last two terms vanish because the control variables are constrained to be a constant, either 0 or 1, therefore:

$$\partial\Psi/\partial X_i = 0 \qquad \text{(eqn 3.32)}$$

and

$$\partial\Phi/\partial X_j = 0 \qquad \text{(eqn 3.33)}$$

35

There are no modifications to the second term of Equation 3.31, but there is a simpler expression for the first term. It is:

$$\sum_i(\partial^2 V(X)/\partial X_i \partial X_j)*dX_i/dt \qquad \text{(eqn 3.34)}$$

$$= \sum_i \partial/\partial X_i(\partial V(X)/\partial X_i)*dX_i/dt$$

$$= d/dt(\partial V(X)/\partial X_j)$$

Equation 3.31 can now be written:

$$d/dt(\partial V(X)/\partial X_j) = -\sum_i(\partial V(X)/\partial X_i) \qquad \text{(eqn 3.35)}$$

$$*\partial/\partial X_j(dX_i/dt)$$

When the X in the denominator of the differential operator of Equation 3.35 is replaced by X_i and Y_j, Equation 3.35 becomes two equations, one for dV_i/dt and another for dW_j/dt. The equations are:

$$dV_i/dt = -\sum_{k=1}^{m} V_k*(\partial/\partial X_i(dX_k/dt)) - \sum_{j=1}^{n} W_j \qquad \text{(eqn 3.36)}$$

$$*\partial/\partial X_i(dY_j/dt)$$

$$dW_j/dt = -\sum_{i=1}^{m} V_i*(\partial/\partial Y_j(dX_i/dt)) - \sum_{l=1}^{n} W_l \qquad \text{(eqn 3.37)}$$

$$*\partial/\partial Y_j(dY_l/dt)$$

In Equation 3.36, the term:

$$\partial/\partial X_i(dX_k/dt) \qquad \text{(eqn 3.38)}$$

simplifies to :

$$\partial/\partial X_i(dX_k/dt) = \begin{cases} 0, & \text{if } i \neq k \\ -\sum_j \beta_{jk}*\psi_{jk}*Y_j, & \text{if } i = k. \end{cases} \qquad \text{(eqn 3.39)}$$

36

Since $\psi_{ji} = 0$ except for the $j^* = j$ that resulted in a minimum for $V_i{}^*\beta_{ji}{}^*X_i$, this term reduces to:

$$\partial/\partial X_i(dX_k/dt) = -\beta_{j*k}{}^*\psi_{j*k}{}^*Y_{j*}, \text{ for } i = k. \qquad \text{(eqn 3.40)}$$

The other second derivative in Equation 3.36 can also be simplified:

$$\partial/\partial X_i(dY_j/dt) = -\alpha_{ij}{}^*\varphi_{ij}{}^*Y_j \qquad \text{(eqn 3.41)}$$

In Equation 3.37, the two second derivative terms can also be simplified:

$$\partial/\partial Y_j(dX_i/dt) = -\beta_{ji}{}^*\psi_{ji}{}^*X_i \qquad \text{(eqn 3.42)}$$

and:

$$\partial/\partial Y_j(dY_l/dt) = \begin{cases} 0, \text{ if } j \neq l \\ -\sum_i \alpha_{il}{}^*\varphi_{il}{}^*X_i, \text{ for } j = l \end{cases} \qquad \text{(eqn 3.43)}$$

As before, $\varphi_{ij} = 0$ for all i except $i^* = i$ that results in a maximum for $W_j{}^*\alpha_{ij}{}^*Y_j$, and the second derivative is then reduced to:

$$\partial/\partial Y_j(dY_l/dt) = -\alpha_{i*l}{}^*\varphi_{i*l}{}^*X_{i*}, \text{ for } j = l. \qquad \text{(eqn 3.44)}$$

With these reduced expressions substituted into Equations 3.36 and 3.37, and with the previously defined Kinematic equations, there are $2(m+n)$ differential equations that describe both the Value of the game to units and the state of each unit as time advances. These equations, with the initial values of the $2(m+n)$ variables, can be solved simultaneously to find the formal solution of the differential game and the other information the commander needs to make the allocation decisions. [Ref. 8:pp. 102-103].

37

7. The Initial Conditions

Differential games are very similar to dynamic programs. If the differential equations are approximated by discrete values in very small sub-intervals of the paths the equations follow, the result would be a discrete game that could be solved by a dynamic program. The values of the state variables X_i and Y_j are known at the start, and Y_j is known at the end. The values of V_i and W_j are not known at the start, but it is possible to find them at the end. If the algorithm started at t_0, it is theoretically possible to enumerate every combination of strategies and resultant variable values, but computationally impractical. As with its discrete cousin, dynamic programming, the approach to solving the differential game is to begin at the end, at time t_f. [Ref. 7:p. 81].

An adjustment that is necessary to start the solution at the end, or the terminal condition, is to reverse time. When moving from the terminal surface toward the initial surface, the symbol τ is used to denote the time interval from the terminal surface to the current position. Figure 3.3 demonstrates this.
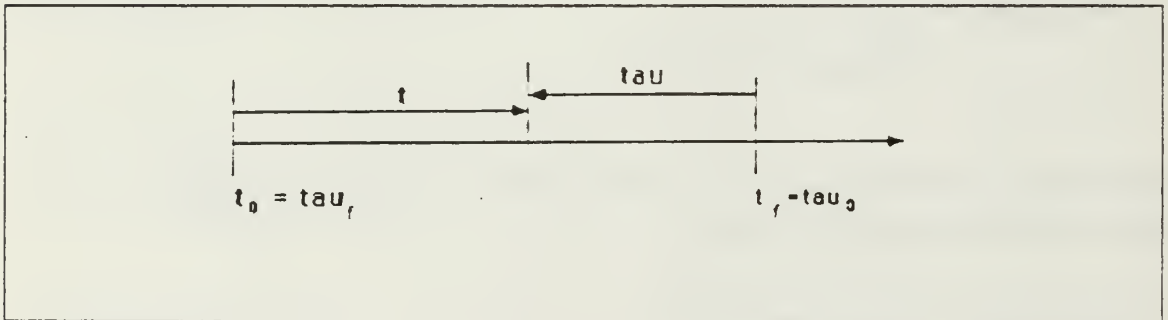
Figure 3.3   Time Scale.

The path equations, derivatives with respect to t, must also be adjusted. They are modified by changing their sign, as in:

$$dXi/d\tau = -dXi/dt \qquad\qquad (eqn\ 3.45)$$

The final preparation is to specify the initial value of the variables, but now the initial value means the value at the terminal surface.

The values of the state variables Y_j are known. They are determined by the inputs specified by the force commander. The values of X_i on the terminal surface are

38

unknown, but they can be any arbitrary positive value. The asset commander is interested in minimizing the power expended, so the quantity of interest is:

$$X_i(t_0)-X_i(t_f) \qquad \text{(eqn 3.46)}$$

the amount of power expended. This quantity can be determined at the end of the game.

The values of V_i and W_j at t_f must be determined. They can be found based on the nature of the game and the given information. The term:

$$V_i = \partial V/\partial X_i \qquad \text{(eqn 3.47)}$$

evaluated at the terminal condition, $t = t_f$ (or $\tau = 0$), equals 0. In other words, at the terminal surface, when Blue has either achieved the decrease in Red's power or has run out of time to achieve it, there is no change in the Value of the game if another increment of Blue power available.

The term:

$$W_j = \partial V/\partial Y_j \qquad \text{(eqn 3.48)}$$

evaluated at the terminal condition equals:

$$-1/\sum_i(\alpha_{ij}*X_i*Y_j*\varphi_{ij}) \qquad \text{(eqn 3.49)}$$

The presence of an additional increment of Y_j at the end of the game translates to a change in the amount of time required for Blue to achieve the required attrition of Red power. Since the Value of the game is a function of this amount of time, it will be directly altered by the presence of the additional increment of Y_j. The reciprocal of the Lanchester attrition equation dY_j/dt is the change in time with respect to Y_j, and the Value of the game is the amount of time required for Blue to decrease Red power, so this reciprocal is the value of $W_j(0)$.

IV. APPLICATION OF THE ALLOCATION ALGORITHM

The motivation to develop the algorithm is to provide an allocation process for Field Artillery battalions in ALARM, so it will be demonstrated in that context. The general scenario will be explained and the allocations determined by the algorithm will be analyzed in this chapter.

A. THE GENERAL SCENARIO

The scenario has a U.S. brigade defending against an attacking WARSAW PACT motorized rifle division. The brigade, the Blue force, will be able to defend successfully if it can maintain a power ratio of 1:3 with the WARSAW PACT division, the Red force. The forecast of the power curves of the Blue brigade and the Red division is shown in Figure 4.1. The power curve of the Red division crosses the power curve of the Blue brigade at time 60. The power curve of the brigade has already been multiplied by three (3) to account for the ratio being considered. After time 60 the brigade will be in an infeasible situation.

The brigade, a component of a Blue division, must receive support from the division in the form of additional combat power if the 1:3 ratio is to be maintained. The Blue division commander could deal with this situation in a number of ways, but in this example he must maintain his present defense and is considering providing additional combat power to the brigade commander. The additional power can come from several different types of units, and the division commander must decide which unit or combination of units to employ. At this point the brigade has an artillery battalion in direct support to it. The division commander wants to know if the remainder of the divisional artillery that is uncommitted to missions of direct support can decrease the power of the Red division by attacking it before time 60 and thus prevent the power curves from crossing until time 65. Time 65 is the limit of the forecast since it is the end of the brigade's area of interest. The asset commander, the division artillery commander, will provide the answer to the explicit question and will also answer the implied questions of what uncommitted units should be told to support the brigade and what missions these units should be given.

The first part of the example will continue to develop the scenario by detailing the information the algorithm uses in the example. The second part will step through the algorithm, with explanations of the allocation scheme given by the algorithm.
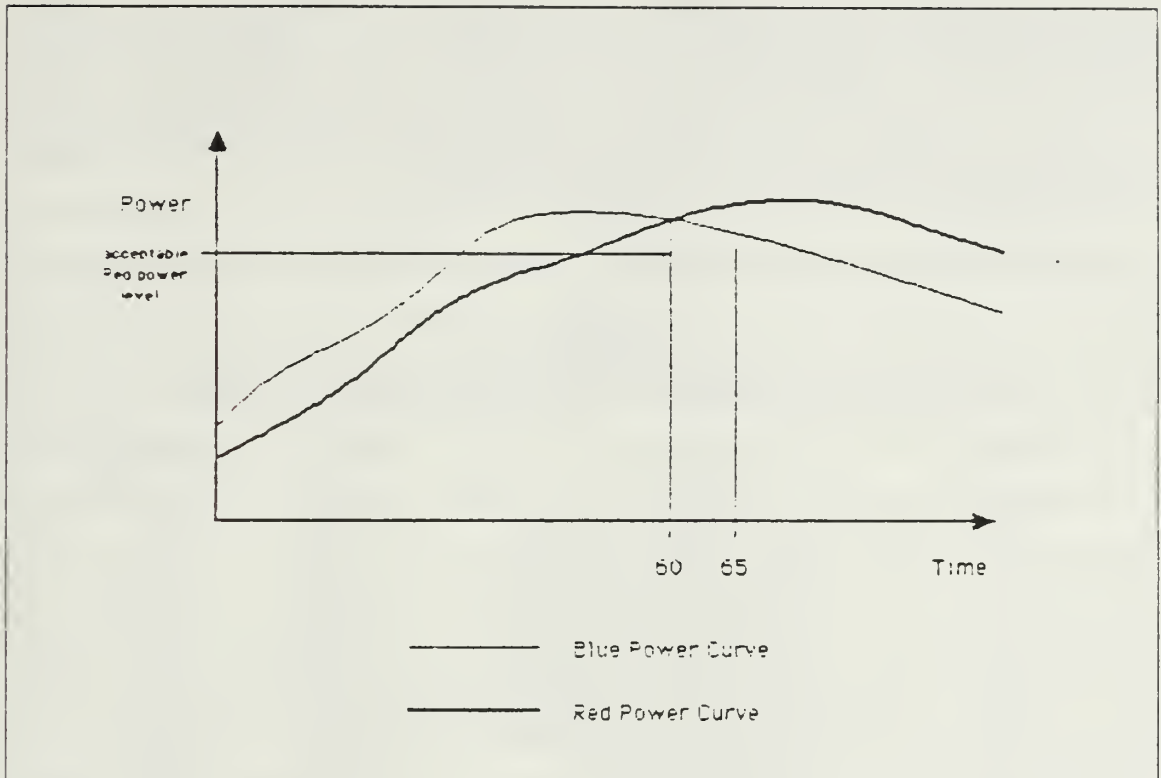
Figure 4.1   Blue Brigade and Red Division Power Curves.

B.   EXPLICIT INPUT VALUES

The time available for the asset commander to decrease the power of the Red division is the time beginning immediately and ending when the Red division power curve crosses the Blue brigade power curve. This time interval could be further reduced by the amount of time necessary to notify his units to begin engaging the Red targets. In this example it is assumed that the Blue units are in position and can begin to engage the Red targets as soon as the allocation scheme is determined. There is no delay for notification. The term "timeframe" will be used in the example to denote the time from t_0 to t_f. Timeframe will also be used in the retrogressive sense. Since the differential game is solved from the terminal condition, where $t = t_f$ and $\tau = 0$, the algorithm times will actually be the retrogressive time, τ.

The uncommitted assets available to the Blue force are two Field Artillery battalions, X_1 and X_2. X_1 has eighteen (18) howitzers and X_2 has twenty-four (24). Five potential targets have been identified in the Red division. Three of them, Y_1, Y_2, and Y_3, are motorized rifle regiments. Y_1 and Y_2 have been fighting the Blue covering

41

force and the forward defensive battalions. Y_3 and Y_4, a tank regiment, are in the Red second echelon and are approaching the defensive line. Y_5 is a logistics unit bringing supplies to Y_1 and Y_2. When these units are resupplied, they will regain the power lost in the initial engagements. The power of the Red and Blue units at the present time and at the end of the time period, t_f, are given in Table I.

TABLE I

SITUATIONAL INHERENT POWER FOR RED AND BLUE

Entity	Entity Type	$SIP(t_0)$	$SIP(t_f|t_0)$
X_1	Blue artillery battalion	1500	400
X_2	Blue artillery battalion	1800	600
Y_1	Red motorized rifle regt.	1800	2970
Y_2	Red motorized rifle regt.	2200	3696
Y_3	Red motorized rifle regt.	3600	6188
Y_4	Red tank regiment	3500	6006
Y_5	Red logistics unit	1200	2059

The SIP of the Blue units at t_f is the minimum power level for those units that is acceptable at the end of the timeframe to the asset commander. The SIP of the Red units at t_f is the power those units will possess if they are unopposed until that time. This is found using the forecasting methods of the Generalized Value System.

The Blue division commander has determined that to maintain the defense, the power of the Red division must be decreased at t_f by 2000 STAPOWs. The sum of the power of the units of the Red force is the total Red power. The assumption that the power of the parent unit is the sum of power of its components does not account for any synergistic forces. These could be included if a form for their representation is found that satisfies the user. The amount of the total power decrease, 2000 STAPOWs, is to be distributed among the component units of the Red force.

The method for distributing the power decrease can take several forms. The primary criteria for selecting a distribution method is that it must closely approximate

42

the power losses that will be assessed by the differential game. If the method used distributes the losses in some manner that does not approximate the outcome of the game, the resulting power levels for the Red units will not conform to the actual situation as they approach the initial surface. The method used in this example begins by determining for each Blue unit the best target at $\tau = 0$ and the best ammunition to fire at that target. The length of time that this ammunition is fired is calculated, based on the rate of fire and the quantity of ammunition available. The power decrease achieved is then calculated by multiplying the Red unit power by the Blue unit power, the attrition coefficient, and the length of time that the ammunition can be fired. If this power decrease is greater than or equal to the amount needed by the force commander, no further attrition of power is needed. If not, the power decrease is subtracted from the Red unit SIP. The SIP's of the other Red units are recalculated for the change in time using the GVS equations. Since the first type of ammunition has been expended, another must be selected for each target type. This changes the attrition coefficients and the best target-firer combinations. With a new ammunition type, the length of time it can be fired must be computed as before, then a power decrease for the best target is computed. The power decrease is added to that previously achieved and the sum is compared with the amount needed. The process is continued iteratively until the power decrease achieved is equal to the amount necessary. As each power decrease is calculated, it is subtracted from the SIP of the Red unit selected. The result is then the power level of the Red unit at $\tau = 0$. The two columns of Table II show the power of the Red units at t_f without the power decrease and with the power decrease applied in the manner described.

The asset commander needs to know where the Red units are located since he will only be able to attack those that are within range of his weapons. In this case, all of the enemy targets are within range of both of the artillery units.

The ammunition available to the artillery units is of four types. For each type, the amount and the attrition coefficient are given in Appendix A. The attrition coefficients are related to the ammunition, not the firing unit. The unit represents the weapon system and the assumption made is that the attrition coefficient is dependent on the ammunition fired, not the weapon system firing it. In this example, the only ammunition available to the units is that which is on hand in the units at t_0.

The ammunition in each unit represents a fraction of that unit's power. In this example, the ammunition in Blue unit X_1 represents fifty percent (50%) of the power

43

TABLE II
POWER OF RED UNITS AT T_F

Red Unit	Forecasted Power	Reduced Power
Y_1	2970	2935
Y_2	3696	3650
Y_3	6188	5019
Y_4	6006	5900
Y_5	2059	1185

of the unit and in X_2 it represents sixty percent (60%). The rate of fire for the weapons system in X_1 is two rounds in three minutes and in X_2 the rate of fire is one round in two minutes.

The determination of the amount of damage the Blue units will suffer in making their attacks requires the assignment of attrition coefficients to the Red units also. These coefficients are shown in Appendix A. The algorithm also needs to know which Red units return fire against the Blue units. A counter is used in the algorithm to count the number of time periods in which a Red unit is selected to be a target by each Blue unit. When the ratio of the counter to the total number of elapsed time periods exceeds a given constant, in this example 0.4, the Red unit is assumed to have located its attacker. If the Red unit is going to return fire against the Blue unit, it is allowed to do so. If the ratio is less than the constant, the Red unit is prevented from returning fire by the assumption that it has not had sufficient opportunity to acquire the attacker. This device is only used in the example. In the actual implementation in ALARM, the detection of Blue units by Red units will be governed by a detailed subroutine that is part of the thesis of CPT Rob Lindstrom [Ref. 9].

The final items of information needed by the algorithm are the rate of increase/decrease in power of each unit as time advances, and the time t_a that represents, for each unit, the time when it reaches its maximum value. The rate of power increase/decrease is D, the exponential rate required by the Generalized Value System. In the example, since the Blue units are in position and can engage targets, it

44

is assumed that their t_a has already passed. The power of these units can only remain constant or decrease during the timeframe under consideration. If a unit does not engage any targets, its power remains constant. If it attacks targets, its power decreases. The Red units Y_1 and Y_2 have been attacking for an undisclosed period of time and have suffered some power attrition. Their t_a is still in the future so their power will increase as they get closer to t_a and as the Red logistics unit gets closer to them. Red units Y_3 and Y_4 are still approaching the main defense area and have not yet started their attacks, so their t_a is in the future. Red unit Y_5 is also still approaching and has not yet reached its t_a. Table III gives the values for power rates and values of t_a.

TABLE III
VALUES OF T_A AND RATE OF POWER INCREASE/DECREASE

Unit	t_a	Rate of Power Change
X_1	Past	.0000
X_2	Past	.0000
Y_1	65	.00835
Y_2	65	.00865
Y_3	65	.009
Y_4	65	.009
Y_5	65	.009

Rate of Power Change is in units of hour^{-1}.

C. ALGORITHM STEPS

The time step used in the example is one hour. Care must be taken to adjust all rate parameters to this scale. The first step in the algorithm is to make feasibility checks on range to targets and on ammunition. In this case, all five Red units are within range for both Blue units. The rate of fire for X_1 and X_2 is .667 and .5 rounds per minute, respectively. Multiplying this by the number of howitzers in each unit

45

yields the result that both units must have seven hundred twenty (720) rounds of an ammunition type for that type to be a candidate for firing in the next hour. There is enough of every type ammunition except type four (4) in both Blue units (Appendix A, Table VIII).

The next step in the algorithm is to select the attrition coefficient the unit will use in the next time step. Having determined the candidate ammunition types in the preceding step, the algorithm now sorts the attrition coefficients of the candidate ammunition types and selects the largest one for each firer-target combination. The result is a vector of five attrition coefficients for each Blue unit. Recall that these steps are not necessary for Red units, which already have an attrition coefficients for each Blue unit. The attrition vectors for Blue firers versus Red units are shown in Table IV for $\tau = 0$.

TABLE IV
ATTRITION COEFFICIENT VECTORS

Units	Targets				
	1	2	3	4	5
X_1	.00009	.00009	.00009	.00005	.000085
X_2	.00009	.00009	.00009	.00005	.000085

Attrition Coefficient Units are Red STAPOW/(Red STAPOW present)(Blue STAPOW)(hour)

The next step is to determine the optimal firer-target combinations for the next hour. In Chapter Three, Section B, Subsection Five, the formula for the Main Equation led to the criteria for optimal combinations. For each Blue unit i and for all Red units j, every possible product $W_j * \alpha_{ij} * Y_j$ is formed and the j that results in the maximum value is the target for the next time step. For each Red unit j and all Blue units i, every possible product $V_i * \beta_{ji} * X_i$ is formed and the i that results in the minimum value is the next target. Thus the optimal pairing of firers and targets for each side of the battle is made. The optimal firer-target combinations for the start of the example are as shown in Table V.

TABLE V
OPTIMAL FIRER-TARGET COMBINATIONS

Firer	Target
X_1	Y_3
X_2	Y_3
Y_1	X_1
Y_2	X_1
Y_3	X_1
Y_4	X_1
Y_5	X_1

Selection of the target for each Blue unit provides the necessary information to determine the power lost due ammunition expenditure. When the target is selected the actual attrition coefficient for the next hour is specified. The attrition coefficient is related to the ammunition to be fired so the type of ammunition is selected. Multiplying the total ammunition power by the ratio of selected ammunition quantity to total ammunition quantity, then dividing by the quantity of selected ammunition (with a user-selected weighting factor for ammunition importance included) results in the power of one round of the selected type. This is multiplied by the number of rounds to be fired by the unit in one hour to determine the power loss due to ammunition expenditure. The power loss due to ammunition expenditure in one hour is 10.58 STAPOWs for X_1 and 16.61 STAPOWs for X_2.

With the information from the preceding steps, the differential game portion of the algorithm is solvable. There are fourteen simultaneous differential equations that are solved using the Subroutine DGEAR from the International Mathematics and Scientific Library (IMSL).

The output from the differential game is in Table VI. It shows that the Blue units need seven (7) hours to achieve the desired power decrease. The attacks start at the fifty-third hour ($t = 53$ or $\tau = 7$). Both Blue units fire on Red unit Y_5, the logistics unit. Blue unit X_2 shifts its fire at 57.352 hours to Red unit Y_3. Blue unit X_1 continues

47

firing on Y_5 until 57.888 hours, then shifts its fire to Y_3. The power levels of the Blue and Red units and the changes in their power levels during the attack are listed in Table VI.

TABLE VI
FIRER-TARGET SOLUTION

Firer	Target	Start Fire(t)	Shift Fire(t)	Shift To
X_1	Y_5	53.00	57.888	Y_3
X_2	Y_5	53.00	57.352	Y_3
X_1	Y_3	57.888	60.00	Cease
X_2	Y_3	57.352	60.00	Cease

Unit	SIP(53)	SIP(60)	ΔPower
X_1	777.610	400.0	377.61
X_2	1042.342	600.0	442.342
Y_3	6148.715	5019.0	1129.715
Y_5	1945.717	1185.0	760.717

The asset commander now has a solution to allocating his resources against the enemy targets. The asset commander decides to assign tactical missions to X_1 and X_2 based on the amount of time needed to achieve the attrition. The units required seven out of a possible sixty hours to accomplish the goal and he recommends that the units be given a General Support mission. If, for example, they had needed forty out of the sixty hours to reach the goal, he might recommend a mission assignment of Reinforcing.

The power levels of Y_3 and Y_5 at the time the attacks begin are critical to the process. The method used to determine their starting values for the retrogressive solver only yields an approximate answer. Working backwards to find the optimal allocations

requires a rule for stopping the attrition when the power levels of the targets are within an acceptable ε of the levels that would be achieved without the attrition. In the example, the attrition is stopped when the sum of the power levels with attrition is within four hundred (400) STAPOWs of the sum of the power levels without attrition.

In this example, the selection of the firer-target combinations is the same for both Blue units. This is a predictable outcome because the Blue units are very similar. The units have exactly the same type of ammunition, as indicated by the ammunition attrition coefficients in Appendix A, Table IX. This coefficient becomes one of the terms in the Equation 3.28 , which is used to select the best target for each firer as explained in Chapter 3, Section B, Subsection 5. The coefficients change when an ammunition type is expended and a new type must be fired. In this case X_2 fires on Y_3 for a longer time than X_1 because there is more of ammunition type three available to X_2. When those two ammunition types are expended the Blue units find better results with their remaining ammunition attacking Y_5.

The example demonstrates that the algorithm provides a workable solution to the allocation problem. The formulation of the scenario led to a predictable outcome. The use of very similar ammunition types and amounts resulted in allocations that were essentially equal. Target selection is intuitively satisfactory. The rule used to select the initial power level of the Red units to solve the differential game was tested in another case and worked equally well there. The solution to the algorithm can be applied to the problem of determining mission assignments or to a more basic scheduling of firers against targets.

V. CONCLUSIONS AND FUTURE DIRECTIONS

An algorithm for solving a problem of continuing military importance has been developed. The allocation process for asset-target combinations was analyzed. The motivations for initiating a decisionmaking process were reviewed, and the goal of the decisionmaker was postulated. The Field Artillery was chosen as an asset for consideration, and the factors that influence the selection of asset-target combinations in this particular functional area were considered. An algorithm was formulated based on these factors and the method of differential games was selected as the optimizing method in the algorithm. The parameters and equations for solving the differential game were developed and the output from the game was used to make mission assignments. Finally, an example was formulated and executed using the algorithm.

The example shows that the algorithm is capable of solving the allocation problem in the type of scenario postulated. Further evaluation of the results using a broad range of situations is necessary to establish full confidence in the algorithm. The algorithm should apply equally well to other asset types, such as attack helicopters or ground support aircraft. These are other variations that should be explored.

The algorithm seems very sensitive to the value of the state variables and the values of other constants. It was observed during test runs that a change of one unit of power in a Red unit was sometimes enough to alter the solution significantly. If this continues to be true, the points were this occurs should be identified. There may be inherent properties of the Kinematic or Path equations that are not yet known that cause this effect. It is very possible that the combinations of these functions lead to irregular surfaces that may be discontinuous at some points. Further use may also require that the technique be implemented with control variables having values between zero and one, not just those two points.

The Value of the differential game in the algorithm was selected to be the time required to decrease Red's power. The Value is an expression dependent upon the decisionmakers being modeled. The expression used in the algorithm may not reflect the goals of every decisionmaker. In other situations the goal of the Blue force commander might be to maximize the decrease in Red's power in a given timeframe. A new expression for the Value of the game would need to be developed in such a case because clearly time is a constraint, not a part of the objective function.

50

Algorithms using differential games seem to be very applicable in the ALARM model because of their ability to treat time continuously and to treat both players as rational decisionmakers. Value expressions can be developed for generic situations or can be designed for specific asset planning. Their application to solving allocations of artillery, close air support, and similar assets in the planning stages of the model appears to be certain.

APPENDIX A
ATTRTION COEFFICIENTS

TABLE VII
AMMUNITION QUANTITIES ON HAND

Ammunition Type	On Hand X_1	X_2
1	8000	8400
2	4000	4800
3	1000	2000
4	600	400

TABLE VIII
AMMUNITION ATTRITION COEFFICIENTS

Ammunition Type	Attrition Coefficient vs Red Unit 1	2	3	4	5
1	.00002	.00004	.000015	.00005	.00007
2	.000065	.00005	.000025	.000045	.000085
3	.00009	.000085	.00009	.00002	.00004
4	.000082	.00009	.000075	.00003	.000055

TABLE IX
ATTRITION COEFFICIENTS FOR RED UNITS

Red Unit	Attrition Coefficient vs Blue Units	
	X_1	X_2
Y_1	.00004	.00003
Y_2	.00003	.00004
Y_3	.00003	.000015
Y_4	.00004	.000045
Y_5	.000001	.00001

APPENDIX B
DEFINITIONS OF POWER TERMS

The Basic Inherent Power ($BIP(X_i)$) is the inherent power possessed by an entity X_i at full strength, when it is in position to engage its most likely adversary as a direct result of X_i's ability to conduct combat operations.

The Adjusted Basic Inherent Power ($ABIP(SX_i(t))$) of an entity X_i at time, t, is the BIP of X_i adjusted for the specific mission and condition of the entity at time t.

The Predicted Adjusted Basic Inherent Power ($PABIP(X_i(t)|SX_i(T_p))$) of an entity X_i at time, t_p, is the ABIP that X_i is predicted to have at time, t ($t > t_p$).

APPENDIX C
FORTRAN 77 CODE FOR ALGORITHM

˙ Variable Definitions:

    N: The number of differential equations to be solved.

    $Y(1), Y(2)$: The Blue force state variables (SIP).

    $Y(3)$-$Y(7)$: The Red force state variables (SIP).

    $Y(8), Y(9)$: The derivatives of the Value function w.r.t. the Blue
                     state variables.

    $Y(10)$-$Y(14)$: The derivatives of the Value function w.r.t. the Red
                     state variables.

    $YP(3)$-$YP(7)$: The PABIP of the Red force units.

    $UNITRT(i)$: The rate of fire for Blue unit i.

    $OH(i,k)$: The quantity of ammunition type k available in Blue unit i

    $TOH(i)$: The total quantity of ammunition available in Blue unit i.

    $NTUBE(i)$: The total number of weapons in Blue unit i.

    $TKIL(j,k)$: The attrition of Red unit j caused by ammunition type k.

    $BETA(j,i)$: The attrition of Blue unit i caused by Red unit j.

    $DR(j)$: The GVS rate of power change of Red unit j.

    $TA(l)$: The time that a unit will reach its maximum SIP.

The first two Commons, DBAND and GEAR, are needed when using DGEAR as the differential equation solver.

```
COMMON/DBAND/NLC,NUC
COMMON/GEAR/DUMMY(48),SDUMMY(4),IDUMMY(38)
```

The variables listed in the COMMON statements that have not been defined are defined in the subroutines.

```
COMMON /ONE/BETA(5,2),PSI(5,2),PHI(2,5),IFLAG(1),TCHG(1),KFLAG(1)
COMMON /TWO/ ISTAR(5), JSTAR(2),YP(7)
COMMON/THREE/UNITRT(2),OH(2,4),TKIL(5,4),TAP(2),TOH(2),NTUBE(2),
1DTIM(1)
COMMON/FOUR/ALFA(2,5),IBEST(2,5),A(2,4),C(2)
COMMON/FIVE/DR(5),TA(1)
COMMON/SIX/DETIND(5,2),JFLAG(1),TPHI(5,2),DFLAG(1)
```

IWK and WK are vectors needed for DGEAR.

```
DIMENSION IWK(20),WK(400)
REAL Y(14),YPRYM(14),TA
```

The EXTERNAL command is required for DGEAR.

```
        EXTERNAL PRYM,PPRYM
        OPEN(UNIT=8,FILE='D1')
        OPEN(UNIT=9,FILE='D2')
        OPEN(UNIT=2,FILE='BEST')
        OPEN(UNIT=3,FILE='AMMO')
        IFLAG(1)=0
        JFLAG(1)=0
        DFLAG(1)=0.
C
C Y(1) AND Y(2) ARE THE SIP FOR X1 AND X2. Y(3) THROUGH Y(7) ARE THE
C SIP FOR Y1 THROUGH Y5. Y(8) AND Y(9) ARE THE VALUES OF V1 AND V2.
C Y(10) THROUGH Y(14) ARE THE VALUES FOR W1 THROUGH W5.
C
        N=14
        Y(1)=400.
        Y(2)= 600.
        Y(3)=2935.
        Y(4)=3650.
        Y(5)=5019.
        Y(6)=5900.
        Y(7)=1185.
        Y(8)=-0.00001
        Y(9)=-0.00001
        Y(10)=WJ(Y(1),Y(2),.00009,.00009,Y(3))
        Y(11)=WJ(Y(1),Y(2),.00009,.00009,Y(4))
        Y(12)=WJ(Y(1),Y(2),.00009,.00009,Y(5))
        Y(13)=WJ(Y(1),Y(2),.000055,.000055,Y(6))
        Y(14)=WJ(Y(1),Y(2),.000085,.000085,Y(7))
C
C THE VALUES YP(I) ARE THE PABIP USED FROM THE GVS METHODOLOGY, BUT THEY
C ARE CONSIDERED TO BE CONSTANT (LIKE AN ABIP).
C
        YP(3)=3097.3
        YP(4)=3860.15
        YP(5)=6461.9
        YP(6)=6282.46
        YP(7)=2153.9
C
        UNITRT(1)=.667
        UNITRT(2)=.5

        OH(1,1)=8000.
        OH(1,2)=4000.
        OH(1,3)=1000.
        OH(1,4)=600.
        OH(2,1)=8400.
        OH(2,2)=4800.
        OH(2,3)=2000.
        OH(2,4)=400.

        TOH(1)=13600.
        TOH(2)=15600.

        NTUBE(1)=18
        NTUBE(2)=24

        TKIL(1,1)=.00002
        TKIL(1,2)=.000065
        TKIL(1,3)=.00009
        TKIL(1,4)=.000082
        TKIL(2,1)=.00004
        TKIL(2,2)=.00005
        TKIL(2,3)=.000085
        TKIL(2,4)=.00009
        TKIL(3,1)=.000035
        TKIL(3,2)=.000025
```

```
TKIL(3,3)=.00009
TKIL(3,4)=.000075
TKIL(4,1)=.00005
TKIL(4,2)=.000055
TKIL(4,3)=.00002
TKIL(4,4)=.00003
TKIL(5,1)=.00007
TKIL(5,2)=.000085
TKIL(5,3)=.00004
TKIL(5,4)=.000055

BETA(1,1)=.00004
BETA(2,1)=.00003
BETA(3,1)=.00003
BETA(4,1)=.00004
BETA(5,1)=.000001
BETA(1,2)=.00003
BETA(2,2)=.00004
BETA(3,2)=.000015
BETA(4,2)=.000045
BETA(5,2)=.00001

DR(1)=.00835
DR(2)=.00865
DR(3)=.009
DR(4)=.009
DR(5)=.009

TA(1)=65.
TIM=0.0
DT=.0001
DTIM(1)=.0001
```

TOL, METH, MITER and INDEX are used by DGEAR only.

```
TOL=.1
METH=1
MITER=3
INDEX=1

CALL FILL(PHI,PSI,TPHI)
JFLAG(1)=1

CALL STAR(ISTAR,JSTAR,Y,TIM)
```

This is the start of the loop that provides solutions to the differential game at every time increment (hour).

```
DO 10 P=1.,60.
```

TAP(i) is the fraction of the total power of Blue unit i credited to the ammunition on hand.

```
TAP(1)=.5*Y(1)
TAP(2)=.6*Y(2)

CALL AMMO(ALFA,IBEST,A)

CALL AMPOW(Y,TIM)

    DO 440 IOLD=1,2
    WRITE(8,450) IOLD,(PHI(IOLD,JOLD),JOLD=1,5)
450 FORMAT(2X,'CURRENT STRATEGY FOR BLUE',I1/,5(F4.2,1X))
440 CONTINUE
```

57

```
      TEND=P
      CALL DGEAR(N,PRYM,PPRYM,TIM,DT,Y,TEND,TOL,METH,MITER,INDEX,IWK,
     1WK,IER)
      IF(IER .GT. 128)PRINT *, 'IER GREATER THAN 128'

      WRITE(8,600) TIM,(Y(I), I=1,14),IDUMMY(7)
  600 FORMAT(1X,F7.4/,7F10.3/,7F10.6/,'NSTEP = ',I5)
```

If the solver has advanced at least one hour, it calls SUBROUTINE DELTA which checks the difference in Red power levels with and without attrition. If the total difference is small enough, it stops the Main.

```
      IF (TIM .GE. 1.0)CALL DELTA(Y,TIM,YP,TDIFF)
      IF(TDIFF .LE. 400.)THEN
          WRITE(8,901) TIM,(Y(LM+2),LM=1,5),TDIFF
  901     FORMAT(1X,'YDELTA ACHIEVED AT TIME ',F6.2/,5(F9.3,2X)/,'TDI
     1FF ',F9.2/)
          GO TO 35
      END IF
```

This is the end of the loop for P.

```
   10 CONTINUE
   35 DO 810 JN=1,5
      DO 820 IN=1,2
      WRITE(8,*)'TPHI', JN,IN, '=', TPHI(JN,IN)
  820 CONTINUE
  810 CONTINUE
      STOP
      END

      SUBROUTINE FILL(A,B,D)
```

This subroutine creates matrices of PHI and PSI filled with zeros.On the first pass it also fills a matrix called TPHI with zeros. This matrix counts the number of times a Blue unit fires on a Red unit and is used for detection and counterfire determination. After the first pass, JFLAG(1) is changed to 1 and TPHI isn't filled again.

PHI is Blue's control variable and PSI is Red's control variable.

```
      COMMON/SIX/DETIND(5,2),JFLAG(1),TPHI(5,2),DFLAG(1)
      REAL A(2,5), B(5,2),D(5,2)
      DO 10 I=1,2
      DO 20 J=1,5
      A(I,J)=0.0
      B(J,I)=0.0
      IF(JFLAG(1) .EQ. 0)D(J,I)=0.0
   20 CONTINUE
   10 CONTINUE
      RETURN
      END


      SUBROUTINE STAR(ISTAR, JSTAR,Y,TIM)
```

This subroutine determines the optimal firer-target combinations. ISTAR (j) is the best target for each Red unit, and JSTAR(i) is the best target for each Blue unit. VLO(j) and WHI(i) are arbitrary values.

```
      COMMON /ONE/BETA(5,2),PSI(5,2),PHI(2,5),IFLAG(1),TCHG(1),KFLAG(1)
      COMMON/FOUR/ALFA(2,5),IBEST(2,5),A(2,4),C(2)
      COMMON/SIX/DETIND(5,2),JFLAG(1),TPHI(5,2),DFLAG(1)
      INTEGER ISTAR(5), JSTAR(2)
      REAL VLO(5),WHI(2),Y(14),MIN,MAX

      IF (IFLAG(1) .EQ. 0)CALL CMPAIR(TIM)
      IF (IFLAG(1) .EQ. 0)CALL AMMO(ALFA,IBEST,A)
```

The first part of the routine finds the best Blue target for each Red firer by finding the smallest value of $V_i * \beta_{ji} * X_i$. PSI(j,i) is set to 1.0 for the best target for each Red unit.

```
      DO 30 J=1,5
      VLO(J)=10000.
      DO 40 I=1,2
      MIN=Y(I+7)*BETA(J,I)*Y(I)
      IF(MIN .LE. VLO(J))THEN
           VLO(J)=MIN
           ISTAR(J)=I
      END IF
   40 CONTINUE
      PSI(J,ISTAR(J))=1.0
   30 CONTINUE
```

The PSI matrix is now determined. To incorporate a detection process in the counterfire Blue receives, the next portion resets a PSI value from 1.0 to 0.0 if the detection indicator (DETIND(j,i) does not exceed an arbitrary value. DETIND(j,i) is computed in SUBROUTINE CMPAIR.

```
      DO 730 J=1,5
      DO 740 I=1,2
      IF(PSI(J,I) .EQ. 1.0 .AND. DETIND(J,I) .LT. .4)PSI(J,I)=0.0
  740 CONTINUE
  730 CONTINUE
```

The next part finds the best target for each Blue firer by finding the largest value of $W_j * \alpha_{ij} * Y_j$. PHI(i,j) is set to 1.0 for the best target for each Blue unit.

```
      DO 50 I=1,2
      WHI(I)=0.000
      DO 60 J=1,5
      MAX=Y(J+9)*ALFA(I,J)*Y(J+2)
      IF(MAX .GE. WHI(I))THEN
           WHI(I)=MAX
           JSTAR(I)=J
      END IF
   60 CONTINUE
```

To keep the program from allocating when a unit is out of ammunition,

the if statement sets PHI to 0 when no ammunition is left.

```
      IF(ALFA(I,JSTAR(I)) .GT. 0.0)THEN
      PHI(I,JSTAR(I))=1.0
      ELSE
         PHI(I,JSTAR(I))=0.0
      END IF
C
   50 CONTINUE
      IFLAG(1)=ISTAR(J)
```

59

```fortran
      RETURN
      END

      SUBROUTINE PRYM(N,TIM,Y,YPRYM)
```

This subroutine is called by DGEAR. It evaluates the derivatives of Y(1)-Y(14)--the Kinematic and Path equations. YPRYM(i) is the derivative of Y(i) with respect to time. It often occurs that DGEAR will call this subroutine multiple times before returning to the Main program, so some of the other subroutines had to be called from PRYM.

```fortran
      DIMENSION  PSUM(14)
      COMMON /ONE/BETA(5,2),PSI(5,2),PHI(2,5),IFLAG(1),TCHG(1),KFLAG(1)
      COMMON /TWO/ ISTAR(5), JSTAR(2),YP(7)
      COMMON/THREE/UNITRT(2),OH(2,4),TKIL(5,4),TAP(2),TOH(2),NTUBE(2),
     1DTIM(1)
      COMMON/FOUR/ALFA(2,5),IBEST(2,5),A(2,4),C(2)
      COMMON/FIVE/DR(5),TA(1)
      COMMON/SIX/DETIND(5,2),JFLAG(1),TPHI(5,2),DFLAG(1)
      REAL YPRYM(14),Y(14)

      CALL FILL(PHI,PSI,TPHI)
      CALL STAR (ISTAR,JSTAR,Y,TIM)
      CALL CMPAIR(TIM)

   79 FORMAT(F9.6/,2(5(F4.2,1X)/))
      DO 120 K=1,N
      IF(K .LE.  2)THEN
          PSUM(K)=0.0
          DO 130 J=1,5
          PSUM(K)=PSUM(K)+(BETA(J,K)*PSI(J,K)*Y(J+2))
  130     CONTINUE
          YPRYM(K)=C(K)+(Y(K)*PSUM(K))
C         WRITE(2,*) TIM,Y(K),PSUM(K)

      ELSE IF(K .LE.  7)THEN
          JA=K-2
          PSUM(K)=0.0
          DO 140 I=1,2
          PSUM(K)=PSUM(K)+(ALFA(I,JA)*PHI(I,JA)*Y(I))
  140     CONTINUE
  199     YPRYM(K)=Y(K)*PSUM(K)-(YP(K)*EXP(-DR(JA)*(TA(1)-TIM))*DR(JA))

      ELSE IF(K .LE.  9)THEN
          PSUM(K)=0.0
          DO 400 I=1,2
          FK=0.0
          DO 410 L=1,5
          FK=FK+(BETA(L,I)*PSI(L,I)*Y(L))
  410     CONTINUE
          PSUM(K)=PSUM(K)+(Y(I+7)*(FK+C(I)))
  400     CONTINUE
          IA=K-7
          YPRYM(K)=-PSUM(K)-(Y(JSTAR(IA)+9)*Y(JSTAR(IA)+2)*ALFA(IA,JSTA
     1    R(IA)))
C     WRITE(2,602) TIM,K,YPRYM(K),PSUM(K),Y(JSTAR(IA)+9),Y(JSTAR(IA)+2),
C    1ALFA(IA,JSTAR(IA))
  602 FORMAT(1X,F6.3,I2/,F10.5,F10.5,F10.5,F10.3,F8.6)

      ELSE
          PSUM(K)=0.0
          DO 420 JZ=1,5
          GL=0.0
```

```
            DO 430 IZ=1,2
            GL=GL+(ALFA(IZ,JZ)*PHI(IZ,JZ)*Y(IZ))
430         CONTINUE
            PSUM(K)=PSUM(K)+(Y(JZ+9)*GL)
420         CONTINUE
            JB=K-9
            YPRYM(K)=-PSUM(K)-(Y(ISTAR(JB)+7)*Y(ISTAR(JB))*BETA(JB,I
     1      STAR(JB)))
         END IF
120 CONTINUE
         RETURN
         END
```

      SUBROUTINE AMPOW(Y,TIM)

This subroutine determines the amount of power expended by Blue units in firing ammunition, $C(i)$. It also counts the ammunition expended and updates the $OH(i,k)$ and $TOH(i)$ quantities and the $TAP(i)$ available.

      REAL Y(14)

$N(i)$ is the number of rounds fired in the time step taken by DGEAR. The time step is generally less than an hour.

```
      INTEGER N(2)
      COMMON/ONE/BETA(5,2),PSI(5,2),PHI(2,5),IFLAG(1),TCHG(1)
      COMMON/TWO/ISTAR(5),JSTAR(2),YP(7)
      COMMON /THREE/ UNITRT(2),OH(2,4),TKIL(5,4),TAP(2),TOH(2),NTUBE(2),
     1DTIM(1)
      COMMON/FOUR/ALFA(2,5),IBEST(2,5),A(2,4),C(2)
      DO 300 I=1,2
      IF(TIM .GT. DTIM(1))THEN
            N(I)=IFIX(UNITRT(I)*NTUBE(I)*(TIM-DTIM(1))*60.)
      ELSE
            N(I)=IFIX(UNITRT(I)*NTUBE(I)*(TIM-IFIX(TIM))*60.)
      END IF
      L=IBEST(I,(JSTAR(I)))
      IF(TOH(I) .GT. 0.0)THEN
            C(I)=A(I,L)*N(I)
            OH(I,L)=OH(I,L)-N(I)
            TOH(I)=TOH(I)-N(I)
            TAP(I)=TAP(I)-C(I)
      END IF
      IF(OH(I,L) .LT. .0001)CALL AMMO(ALFA,IBEST,A)
C     WRITE(3,302) OH(I,L),TOH(I)
C 302 FORMAT(1X,'AMMO ON HAND IS ',F12.5,' TOTAL AMMO ON HAND IS ',F12.5
C    1)
300 CONTINUE
      DTIM(1)=TIM
      RETURN
      END
```

      SUBROUTINE AMMO(ALFA,IBEST,A)

This subroutine passes three items back to the Main. It determines the α matrix, picks the best ammunition type to fire at each possible target, and determines the power of each round of each ammunition type. ALFA is the matrix of attrition coefficients for all firer-target pairs. IBEST is the best ammunition to fire at each possible target. 'A' is the matrix of ammunition power per round in each Blue unit.

61

```
      COMMON/THREE/UNITRT(2),OH(2,4),TKIL(5,4),TAP(2),TOH(2),NTUBE(2),
     1DTIM(1)
```

FIRE(i) is the number of rounds Blue unit i fires in 60 minutes. AVAIL(i,k) is a dummy matrix that is only used in the routine.

```
      REAL ALFA(2,5),A(2,4),AVAIL(2,4),FIRE(2),MAX,KTAP(2,4)
      INTEGER IBEST(2,5)
      DO 200 I=1,2
      FIRE(I)=UNITRT(I)*NTUBE(I)*60.
```

The routine compares the quantity of ammunition on hand (OH) with the quantity the unit will fire in the next hour. If OH is large enough, that type becomes a candidate for firing and the attrition coefficient for round k against target j is passed into AVAIL(I,K).

```
      DO 210 J=1,5
      DO 220 K=1,4
      IF(OH(I,K).LT. FIRE(I))THEN
           AVAIL(I,K)=0.0
      ELSE
           AVAIL(I,K)=TKIL(J,K)
      END IF
  220 CONTINUE
```

Now the ammunition types available in each Blue unit i are sorted to find the one that gives the maximum attrition against Red target j. The attrition coefficient for this one becomes the attrition coefficient for the Blue unit. IBEST(i,j) is determined simultaneously. The routine then loops back to consider the next possible target.

```
      MAX=0.
      DO 230 K=1,4
      IF(AVAIL(I,K) .GT. MAX)THEN
           MAX=AVAIL(I,K)
           IBEST(I,J)=K
      ELSE
           D=0
      END IF
  230 CONTINUE
      ALFA(I,J)=MAX
  210 CONTINUE
  200 CONTINUE
```

The 'A(i,k)' matrix is now created. For each ammunition type, the OH quantity is divided by the TOH quantity. The result is multiplied by the fraction of the unit power represented by ammunition, TAP. This is then the fraction of TAP represented by each ammunition type, KTAP. KTAP is then divided by the OH quantity to get the amount of power in each round. This could be further adjusted for the relative importance of each round type.

```
      DO 240 I=1,2
      DO 250 K=1,4
```

```
        IF (OH(I,K) .GT. .0001)THEN
            KTAP(I,K)=(OH(I,K)/TOH(I))*TAP(I)
            A(I,K)=KTAP(I,K)/OH(I,K)
        ELSE
            A(I,K)=0.0
        END IF
250 CONTINUE
240 CONTINUE
        DO 320 I=1,2
320 CONTINUE
        RETURN
        END

        SUBROUTINE PPRYM(N,TIM,Y,PD)
```

This is a dummy subroutine. It must be used to evaluate the Jacobian matrix in some applications of DGEAR.

```
        INTEGER N
        REAL Y(14),PD(14,14)
        RETURN
        END

        SUBROUTINE CMPAIR(TIM)
```

This subroutine checks the strategy for each Blue unit and reports if the strategy has changed. It also computes the detection indicator DETIND(j,i).

```
        COMMON/ONE/BETA(5,2),PSI(5,2),PHI(2,5),IFLAG(1),TCHG(1),KFLAG(1)
        COMMON/SIX/DETIND(5,2),JFLAG(1),TPHI(5,2),DFLAG(1)
        COMMON/GEAR/DUMMY(48),SDUMMY(4),IDUMMY(38)
```

PAST1(i) is a matrix used as an interchange in this routine only.

```
        DIMENSION PAST1(2,5)
        IF (IFLAG(1) .EQ. 0) GO TO 280
        KFLAG(1)=0

        DO 290 IC=1,2
        DO 330 JC=1,5
        IF (PAST1(IC,JC) .EQ. PHI(IC,JC))GO TO 330
        KFLAG(1)=1
330 CONTINUE
290 CONTINUE
```

If the solver has taken a step forward in time and the strategy for a Blue unit has changed, the subroutine reports it here.

```
        IF(TCHG(1) .NE. TIM)THEN
            IF(KFLAG(1) .NE. 1)GO TO 701
            DO 700 INEW=1,2
            WRITE(8,601) INEW,TCHG,(PHI(INEW,JNEW),JNEW=1,5)
601     FORMAT(2X,'STRATEGY FOR BLUE',I1,' CHANGED AT TIME ',F6.3/,5(F4
    1       .2,1X))
700     CONTINUE
```

The subroutine increments DFLAG(1) because time has advanced, and adds PHI(i,j) to TPHI(i,j). It then determines DETIND(j,i), the ratio of the number of times i engages j to the total potential engagement time.

```
701     DFLAG(1)=DFLAG(1)+1.
        DO 710 JN=1,5
        DO 720 IN=1,2
```

63

```
          TPHI(JN,IN)=TPHI(JN,IN)+PHI(IN,JN)
          DETIND(JN,IN)=TPHI(JN,IN)/DFLAG(1)
720    CONTINUE
710    CONTINUE
       END IF
```

Before exiting the routine, PHI(i,j) values are passed to PAST1(i,j).

```
280 DO 340 ID=1,2
       DO 350 JD=1,5
       PAST1(ID,JD)=PHI(ID,JD)
350 CONTINUE
340 CONTINUE
       TCHG(1)=TIM
       RETURN
       END

       FUNCTION WJ(A,B,C,D,E)
```

This function determines the values of W(j), a.k.a. Y(10)-Y(14).

```
       DEN=(A*C+B*D)*E
       WJ=1./DEN
       RETURN
       END

       SUBROUTINE DELTA(E,F,G,TDIFF)
```

This subroutine measures the difference between the power levels of the Red units with and without attrition. It sums the difference for all of the units and passes this back to the Main.

```
       COMMON /ONE/BETA(5,2),PSI(5,2),PHI(2,5),IFLAG(1),TCHG(1),KFLAG(1)
       COMMON/FIVE/DR(5),TA(1)
       REAL E(14),G(5),DIFF(5),X(5)
       DO 830 M=1,5
```

X(m) is the power level of Red unit m without attrition.

```
       X(M)=G(M+2)*EXP(-DR(M)*(F+5.))
       DIFF(M)=E(M+2)-X(M)
830 CONTINUE
       TDIFF=0.0
       DO 860 L=1,5
       TDIFF=TDIFF+ABS(DIFF(L))
860 CONTINUE
       WRITE(9,  *) TDIFF,F
       WRITE(9,861)(DIFF(M),M=1,5),(X(L),L=1,5)
861 FORMAT(1X,5(F9.3,2X)/,5(F9.3,2X)/)
       RETURN
       END
```

LIST OF REFERENCES

1. Department of the Army Field Manual 100-5, *Operations*, Washington, D.C., October 1982.

2. Hartman, J.K., Parry, S.H. and Schoenstadt, A.L., *AirLand Research Model*, paper presented to MORS, Naval Postgraduate School, Monterey, California, June 1984.

3. Kilmer, Robert, *The Generalized Value System and Future State Decision Making*, M.S. Thesis, Naval Postgraduate School, Monterey, California, March 1986.

4. Taylor, James G., *Lanchester Models of Warfare*, Military Operations Research Society, March 1983.

5. Naval Postgraduate School, NPS-55TW70062A, *Application of Differential Games to Problems of Military Conflict: Tactical Allocation Problems*, by James G. Taylor, 19 June 1970.

6. Department of the Army Field Manual 6-20, *Fire Support in Combined Arms Operations*, Washington, D.C., January 1983.

7. Isaacs, Rufus P., *Differential Games*, John Wiley and Sons, Inc., 1967.

8. Owen, Guillermo, *Game Theory*, Academic Press, 1982.

9. Lindstrom, Rob, *A Field Artillery Module for the Airland Research Model*, M.S. Thesis, Naval Postgraduate School, Monterey, California, June 1986.

BIBLIOGRAPHY

Sternberg, Stanley D., *Development of Optimal Allocation Strategies in Heterogeneous Lanchester-Type Processes*, Ph.D. Dissertation, University of Michigan, 1971.

Weiss, Herbert K.,"Some Differential Games of Tactical Interest and the Value of a Supporting Weapon System," *Operations Research*, v. 7, pp. 180-196, 1959.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center   2
   Cameron Station
   Alexandria, Virginia 22304-6145

2. Library, Code 0142
   Naval Postgraduate School
   Monterey, California 93943-5002

3. Deputy Undersecretary of the Army   2
   for Operations Research
   Room 2E261, Pentagon
   Washington, D.C. 20310

4. Director   1
   U.S. Army TRADOC Analysis Center
   White Sands Missile Range, New Mexico 88002

5. Commander   3
   U.S. Army TRADOC Analysis Center
   Attn: Mr. Reed Davis
   Fort Leavenworth, Kansas 66027

6. Director   1
   Attn: Mr. E. B. Vandiver III
   U.S. Army Concepts Analysis Agency
   Bethesda, Maryland 20814

7. Bell Hall Library   1
   U.S. Army Combined Arms Center
   Fort Leavenworth, Kansas 66027

8. Dr. Samuel H. Parry, Code 55Py   5
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California 93943

9. Dr. Guillermo Owen, Code 53On   5
   Department of Mathematics
   Naval Postgraduate School
   Monterey, California 93943

10. Director   1
    Attn: COL Tony Brinkley
    Studies and Analysis Directorate
    Headquarters, U.S. Army TRADOC
    Fort Monroe, Virginia 23651

11. Captain John M Geddes, Jr.   2
    1235 Limit Street
    Leavenworth, Kansas 66048

12. Lieutenant Seong Pil Hur   1
    Postal Code G02
    Jin-Hae Si Song-Hak Dong 5
    Seoul, Korea

13.   Dept. of Operations Sciences   1
        Attn: MAJ Dan Reyen
        AFIT/ENS
        Wright-Patterson AFB, Ohio 45433

14.   Director   1
        Army Models Management Office (AMMO)
        Attn: COL Ken Wiersima
        U.S. Army Combined Arms Center
        FT Leavenworth, Kansas 66027

15.   MAJ Joseph Manzo   1
        Department of Mathematics
        United States Military Academy
        West Point, New York 10996